

Surface de programmation  
pour dispositifs de la série 300

# **OED3**

**Version 1.0**

Doc. No. 212.954/DF 04.94

No. d'ident.: 00441109640

Edition: a000 Avril 94



**Table des matières**

		<b>Page</b>
<b>1</b>	<b>Description générale</b>	<b>1-1</b>
1.1	Documentations techniques	1-2
1.2	Contenu du programme-produit	1-2
1.3	Application	1-2
1.4	Programmation et maniement d'une commande	1-3
1.5	Modes d'exploitation d'une commande de la série 300 avec OED3	1-5
1.6	Programmation du déplacement	1-8
1.6.1	Modes d'exploitation d'axe	1-8
1.6.1.1	Mode point à point	1-8
1.6.1.2	Mode de poursuite de position	1-11
1.7	Programmation de l'encodeur	1-13
1.8	Interpolation avec unités de positionnement multiaxes	1-16
1.9	Régulation de position pour WDP3-337 et WDP3-338	1-18
1.10	Programmation d'interface	1-19
1.11	Synchronisation avec la commande prioritaire	1-21
1.12	Exploitation avec carte d'entrée/sortie MP 926	1-22
<b>2</b>	<b>Installation</b>	<b>2-1</b>
2.1	Volume de livraison	2-1
2.2	Accessoires	2-1
2.3	Conditions	2-2
2.4	Schéma de connexion	2-2

## **Table des matières**

---

	<b>Page</b>
<b>3 Programmation</b>	<b>3-1</b>
3.1 Lancer OED3	3-1
3.2 Surface de programmation	3-3
3.2.1 Programmes d'édition et fonctions	3-4
3.3 Programme d'édition de paramètres	3-5
3.3.1 Afficher les paramètres	3-5
3.3.2 Editer les paramètres	3-9
3.4 Programme d'édition API	3-10
3.4.1 Liste d'instructions	3-10
3.4.2 Afficher le programme API	3-23
3.4.3 Insérer une ligne	3-23
3.4.4 Effacer ligne(s)	3-24
3.4.5 Editer une ligne	3-24
3.5 Programme d'édition de déroulement	3-25
3.5.1 Liste d'instructions	3-25
3.5.2 Afficher programme de déroulement	3-44
3.5.3 Insérer une ligne	3-44
3.5.4 Effacer ligne(s)	3-45
3.5.5 Editer une ligne	3-45
3.6 Programme d'édition de texte	3-46
3.6.1 Afficher textes	3-46
3.6.2 Editer textes	3-46
3.7 Fonction Upload	3-47
3.8 Fonction Download	3-48
3.9 Terminer le mode d'édition	3-48

<b>4</b>	<b>Messages d'erreur</b>	<b>Page 4-1</b>
<b>5</b>	<b>Annexe</b>	<b>5-1</b>
5.1	Termes techniques	5-1
5.2	Abréviations	5-4
<b>6</b>	<b>Index</b>	<b>6-1</b>

## ***Table des matières***

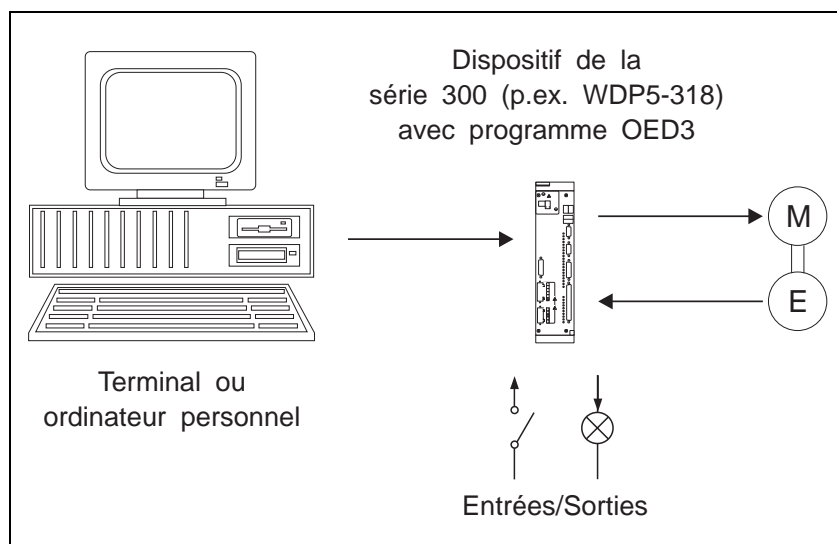
---

## 1 Description générale

La surface de programmation OED3 sert à la génération de programmes utilisateurs simples pour les commandes BERGER LAHR de la série 300 (p.ex. WDP5-318) à l'aide d'un terminal ou ordinateur personnel.

Les fonctions suivantes peuvent être exécutées à l'aide de la surface de programmation OED3:

- Paramétrage et programmation de la commande
- exécution d'un programme utilisateur ligne par ligne (mode pas à pas).
- déroulement automatique d'un programme utilisateur (mode automatique)
- programmation aisée du mode manuel et teach-in grâce aux indicateurs prédéfinis



*Fig. 1-1 Intégration dans le système*

### **1.1 Documentations techniques**

Cette documentation comporte des informations sur l'installation et le maniement de la surface de programmation OED3.

La documentation technique du dispositif comporte des informations qui sont spécifiques à la commande.

### **1.2 Contenu du programme-produit**

Le programme-produit OED3 comporte les fichiers listés ci-après:

- Problèmes-exemples
- Fichier README
- Fichier EXE pour assistance off-line
- Programme de terminal BTERM

### **1.3 Application**

La surface de programmation OED3 sert à la génération de programmes utilisateurs pour les commandes BERGER LAHR de la série 300 (p.ex. WDP5-318) à l'aide d'un terminal ou ordinateur personnel.



## 1.4 Programmation et maniment d'une commande

Le programme OED3 est déjà intégré dans les commandes BERGER LAHR de la série 300.

La commande peut être programmée en mode d'édition à l'aide d'un terminal ou ordinateur personnel (avec un programme de terminal, p.ex. BTERM), se reporter au chapitre 3.

### *Programme utilisateur*

Le programme utilisateur qui a été généré en mode d'édition est exécuté en mode automatique de la commande aussitôt que le mode d'édition a été abandonné.

Un programme utilisateur comporte un programme de déroulement et un programme API qui sont exécutés en parallèle (selon le principe de la tranche de temps) dans la commande.

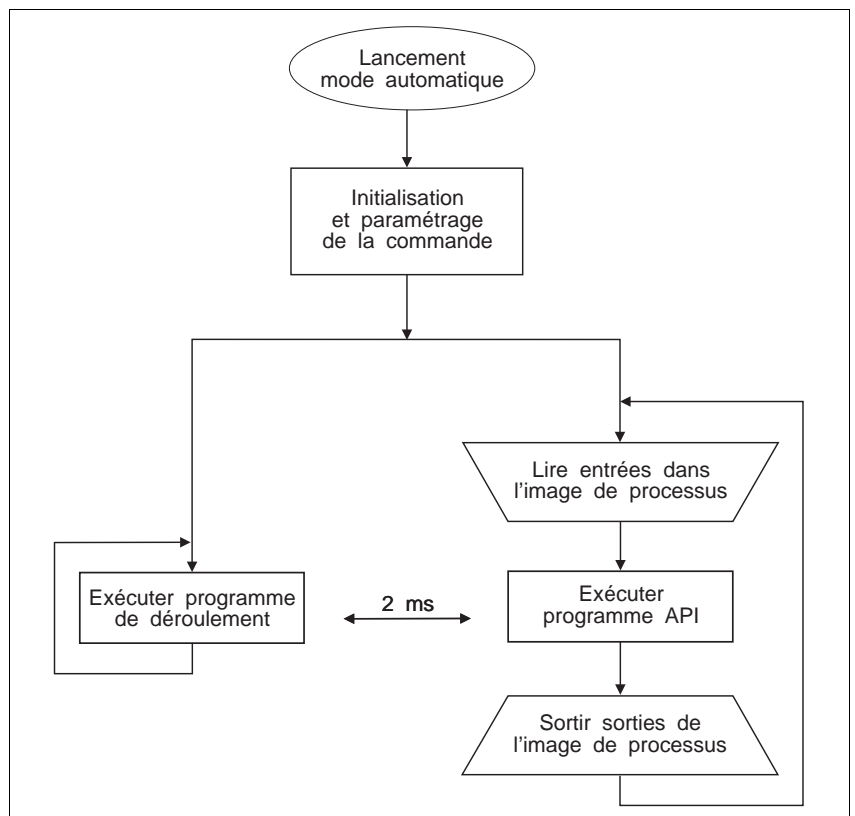


Fig. 1-2 Exécution de programme en mode automatique

## Description générale

---

*Programme API* Les fonctions suivantes peuvent être exécutées au moyen du programme API:

- opérations logiques (ET, OU);
- opérations arithmétiques (addition, soustraction, multiplication, division);
- opérations de transfert (charger, mémoriser);
- opérations de comparaison (égal, supérieur à, inférieur à);
- saut conditionnels et inconditionnels.

Le programme API est exécuté de manière cyclique lorsque la commande est en mode automatique. Après avoir lancé le mode automatique, les états d'entrée sont lus dans l'image de processus (mémoire intermédiaire pour entrées/sorties). Les instructions programmées dans le programme API sont ensuite exécutées et les états de sortie de l'image de processus mis à jour. Finalement, les états de sortie de l'image de processus sont sortis et le cycle recommence (le temps de cycle n'est pas surveillé).

*Programme de déroulement* Le programme de déroulement permet d'exécuter les fonctions suivantes:

- réglage des modes d'exploitation d'axe (mode point à point, mode de poursuite de position);
- exécution de positionnements (absolus et relatifs) et de courses de référence;
- transmission et réception de caractères et variables via l'interface sérielle;
- synchronisation avec la commande prioritaire (au moyen de l'entrée/sortie de synchronisation).

De plus, la plupart des fonctions du programme API peuvent être exécutées. Mais l'accès aux entrées/sorties se réalise directement et non pas au moyen de l'image de processus du programme API.



### NOTE

*La communication entre programme API et programme de déroulement peut se faire à l'aide d'indicateurs ou de variables.*

*Teach-In* Des positions qui ont été programmées dans le programme utilisateur peuvent être approchées et recouvertes en mode teach-in.

*Programme d'édition de texte* Le programme d'édition de texte sert à introduire des chaînes de caractères qui doivent être transmises dans le programme de déroulement.

*Programme d'édition de paramètres* Le programme d'édition de paramètres sert à introduire de différents paramètres pour le programme de déroulement.

## 1.5 Modes d'exploitation d'une commande de la série 300 avec OED3

La version standard de la commande dispose de deux modes d'exploitation:



– Mode utilisateur pour l'exécution d'un programme utilisateur;



– Mode manuel pour le déplacement manuel du moteur au moyen des touches sur la platine frontale.



**NOTE**

Le mode utilisateur et le mode manuel de la commande sont décrits dans la documentation du dispositif.

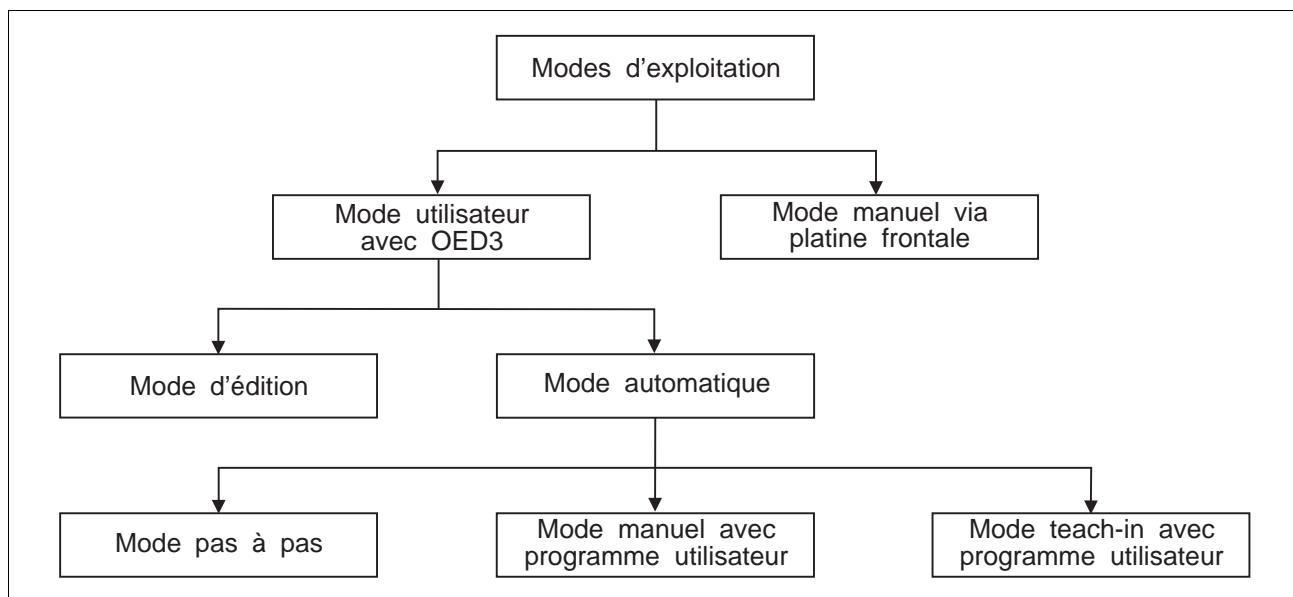
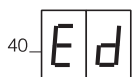


Fig. 1-3 Modes d'exploitation de la commande

Les modes d'exploitation suivants peuvent être sélectionnés en mode utilisateur:

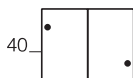
### Mode d'édition



Le mode d'édition est activé en amenant la touche de sélection 41 en position +.

En mode d'édition, un programme utilisateur peut être introduit et affiché ligne par ligne à l'aide d'un terminal ou ordinateur personnel. Lorsqu'un ordinateur personnel (avec programme de terminal, p.ex. BTERM) est utilisé en tant que dispositif de programmation, il est possible de charger un programme utilisateur complet depuis la commande dans l'ordinateur personnel (upload) ou depuis l'ordinateur personnel dans la commande (download), se reporter au chapitre 3.

## Mode automatique



Le programme utilisateur qui comporte un programme API et un programme de déroulement est exécuté en mode automatique. Le mode automatique est activé aussitôt que le mode d'édition (avec <X> et <J>) a été abandonné. Si programmé dans le programme utilisateur, le mode manuel, teach-in ou pas à pas peut être sélectionné en mode automatique, se reporter au chapitre 3.



### NOTE

*Il est possible d'activer ou de désactiver la sortie de test au moyen d'un indicateur lorsque la commande est en mode automatique. Pendant la sortie de test, le programme de déroulement est affiché ligne par ligne sur l'écran.*

Les indicateurs prédéfinis sont listés dans le tableau suivant.

Indicateur	Signification			
m0	Course manuelle à droite			
m1	Course manuelle à gauche			
m2	Course rapide			
m3	Sélection multiaxes bit 0	Axe	m4	m3
m4	Sélection multiaxes bit 1	1	0	0
		2	0	1
		3	1	0
		4	1	1
m5	Activer teach-in			
m6	Prendre position en charge			
m7	Ne pas prendre position en charge			
m8	Réservé			
m9	Sortie de test 0 = arrêt, 1 = marche			
m10	Activer manuel 0 = non activé, 1 = activé			
m11	Course manuelle activée? 0 = non, 1 = oui			
m12	Sortie de test 0 = numéro de ligne, 1 = ligne complète			
m13 à m20	Réservé			
m21 à m1000	Librement disponibles; pour commande avec interface interbus-S:			
m32 à m111	pour les entrées étendues			
m112 à m191	pour les sorties étendues			
m1001 à m1023	Constantes du système			



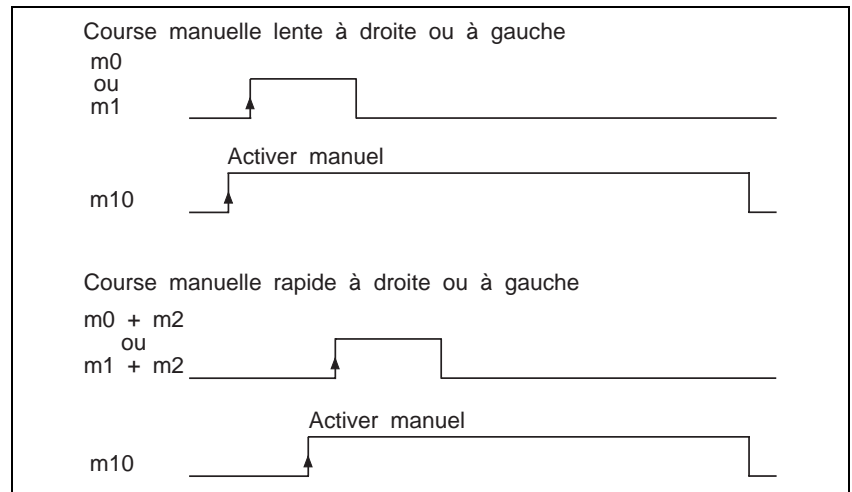
### NOTE

*Vous trouverez des problèmes-exemples pour le mode manuel et teach-in dans le programme-produit OED3 livré.*

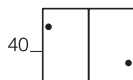
#### – Mode manuel

En mode manuel, il est possible de déplacer l'axe lentement ou rapidement (course rapide) de manière manuelle. S'il s'agit de systèmes multiaxes, l'axe désiré doit être sélectionné avant la course manuelle. L'activation d'une course manuelle ou la sélection d'axe (pour systèmes multiaxes) se réalise dans le programme utilisateur au moyen d'indicateurs prédéfinis, se reporter à la fig. 1-4.

Fig. 1-4 Diagramme des temps pour mode manuel

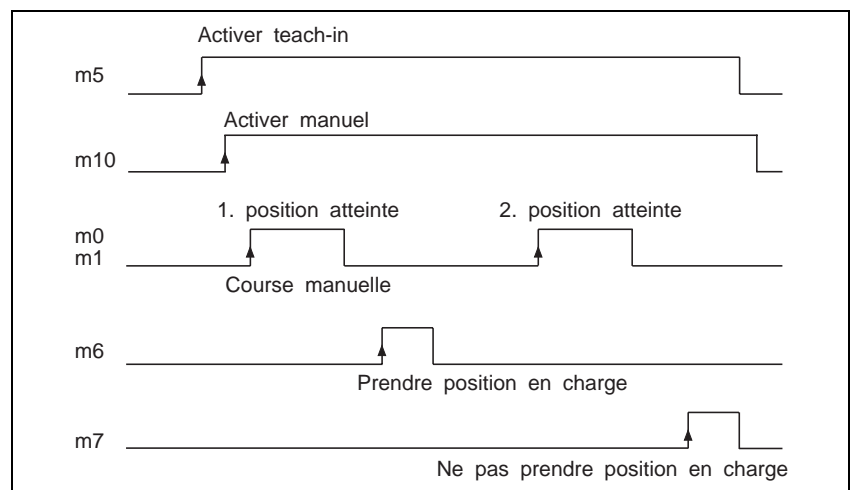


– Mode teach-in



En mode teach-in, il est possible de s'approcher de positions manuellement et de les mémoriser dans le programme de déroulement. Le programme de déroulement est exécuté normalement jusqu'à ce qu'une instruction de positionnement ait été exécutée. Le programme de déroulement s'arrête ensuite et la position programmée peut être modifiée au moyen de teach-in. Ainsi, il est possible de recouvrir les positions programmées du programme de déroulement, l'une après l'autre, par les positions modifiées au moyen de teach-in. Des indicateurs prédéfinis dans le programme utilisateur permettent d'activer le mode teach-in ou de mémoriser ou ignorer des positions, se reporter à la fig. 1-5. Le mode teach-in ne peut être réalisé qu'en mode d'exploitation d'axe point à point.

Fig. 1-5 Diagramme des temps pour mode teach-in



– Mode pas à pas

En mode pas à pas, le programme de déroulement peut être exécuté ligne par ligne (au moyen de la touche ENTER) et affiché sur l'écran. Le programme API continue de se dérouler pendant le mode pas à pas. L'activation du mode pas à pas au moyen du paramètre 34 est décrite dans le chapitre 3.3.1.

### 1.6 Programmation du déplacement

Les chapitres suivants expliquent les concepts et correlations qui sont nécessaires à la programmation du déplacement.

#### 1.6.1 Modes d'exploitation d'axe

L'instruction "mode" permet de sélectionner les modes d'exploitation d'axe suivants dans le programme de déroulement (si l'axe est à l'arrêt):

- Mode point à point
- Mode de poursuite de position

##### 1.6.1.1 Mode point à point

En mode point à point, une instruction de positionnement est utilisée pour le déplacement d'un point A à un point B. Le positionnement peut être absolu (basé sur le point de référence de l'axe) ou relatif (basé sur la position actuelle de l'axe). De plus, il est possible de spécifier si le programme doit attendre jusqu'à ce qu'un positionnement ait été exécuté ou si le programme doit se dérouler en parallèle. Les positions sont indiquées en unités définies par l'utilisateur, p.ex. 100 (mm).

*Facteur de cadrage pour valeurs de position*

Grâce aux paramètres "facteur de cadrage numérateur" et "facteur de cadrage dénominateur", il est possible de convertir des unités définies par l'utilisateur, p.ex. 100 mm, en unités d'entraînement, p.ex. 1000 pas (ou une révolution du moteur).

Il s'applique:

Unités d'entraîn. = unités définies par l'utilisateur x facteur de cadrage

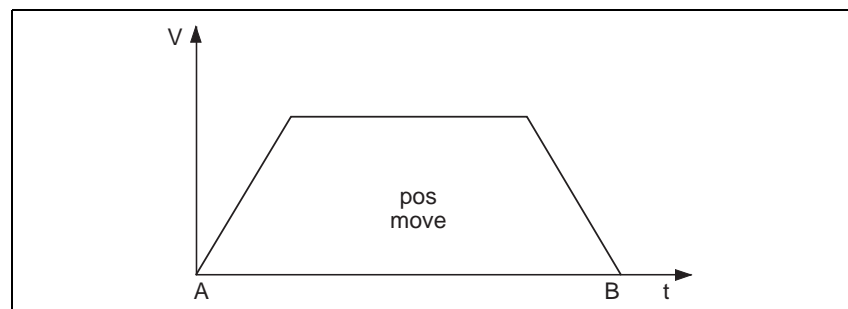


Fig. 1-6 Mode point à point

Paramètres programmables pour mode point à point:

- "Interrupteurs limiteurs actifs"
- "Evaluation d'encodeur" (pour contrôle de rotation)
- "Réglage d'encodeur" (pour contrôle de rotation)
- "Facteur de cadrage dénominateur"
- "Facteur de cadrage numérateur"
- "Rampe"
- "Accélération standard"
- "Vitesse standard"
- "Vitesse start/stop"

Le chapitre 3.3.1 décrit la signification et le pré-réglage des paramètres.

Instructions pour la programmation du déplacement en mode point à point:

Instruction	Signification
"acc"	Sélectionner courbe d'accélération/décélération
"mode"	Régler mode d'exploitation d'axe (mode point à point)
"move"	Positionnement relatif en unités définies par l'utilisateur
"movef"	Positionnement relatif en unités définies par l'utilisateur et attendre jusqu'à ce que position soit atteinte
"pos"	Positionnement absolu en unités définies par l'utilisateur
"posf"	Positionnement absolu en unités définies par l'utilisateur et attendre jusqu'à ce que position soit atteinte
"ref"	Course de référence vers interrupteur limiteur ou interrupteur de référence
"reff"	Course de référence vers interrupteur limiteur et interrupteur de référence et attendre jusqu'à ce que point de référence soit atteint
"stop"	Arrêter déplacement d'axe
"vel"	Régler vitesse exigée



### NOTE

*Pendant un positionnement avec "pos" ou "move", le programme continue d'être exécuté ce qui signifie qu'il est par exemple possible de lire les signaux d'entrée, de positionner les signaux de sortie ou de modifier la distance ou la vitesse pour le positionnement.*

## Course de référence

Des courses de référence peuvent être réalisées au moyen des instructions "ref" et "reff".



### NOTE

Une course de référence ne peut être exécutée qu'en mode point à point.

A l'aide d'une course de référence on s'approche d'un point de référence qui est utilisé en tant que zéro pour le système de coordonnées.

Le paramètre "type de la course de référence" sert à déterminer si l'on s'approche pendant la course de référence de

- l'interrupteur limiteur négatif,
- l'interrupteur limiteur positif ou
- l'interrupteur de référence (rotation à gauche ou rotation à droite).

Le principe de différentes courses de référence est évident dans les figures 1-7 et 1-8.

Une distance de sécurité vers l'interrupteur limiteur ou l'interrupteur de référence peut être programmée au moyen du paramètre "distance de libération".

Fig. 1-7 Principe de la course de référence vers interrupteur limiteur

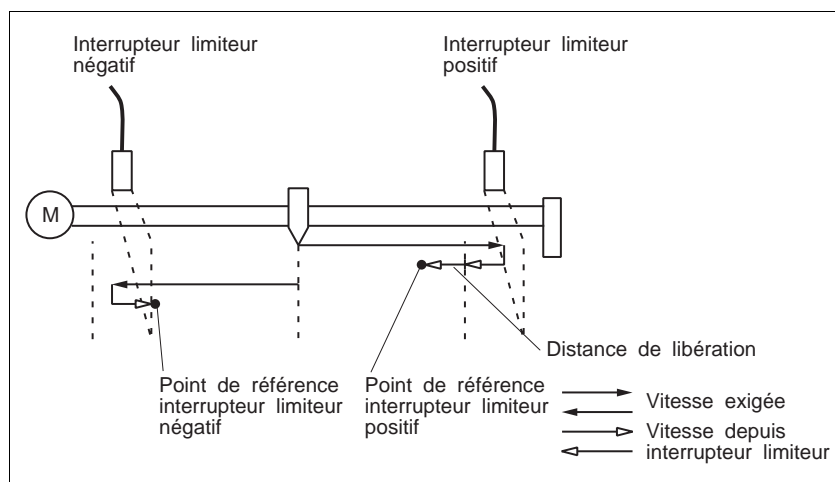
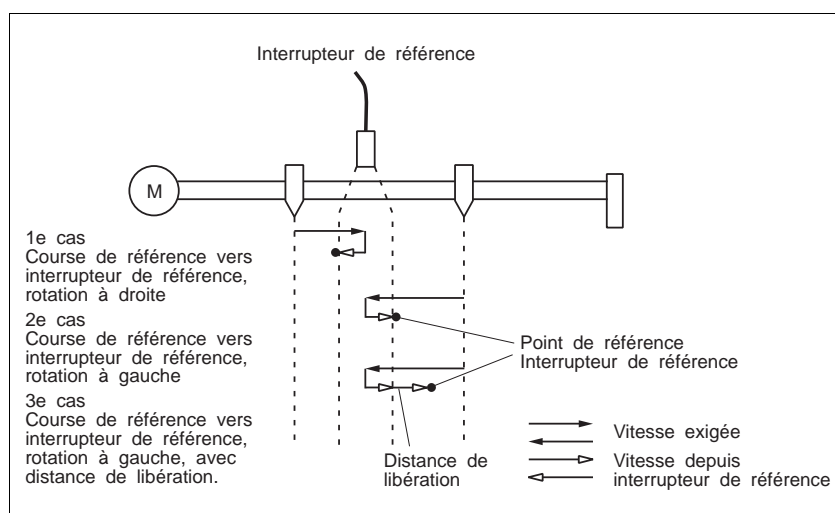


Fig. 1-8 Principe de la course de référence vers interrupteur de référence





**1.6.1.2 Mode de poursuite de position**

En mode de poursuite de position, les positions sont prédéfinies à l'aide d'une connexion d'encodeur ou d'une variable (se reporter également au chapitre 1.7). La position exigée peut être modifiée avec un facteur d'engrenage, ainsi assurant la réalisation d'un engrenage électronique.

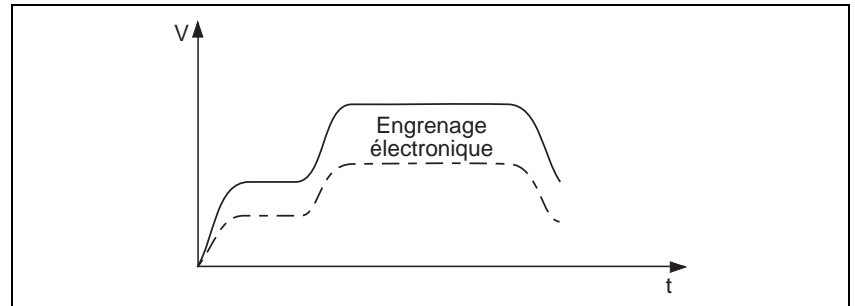


Fig. 1-9 Mode de poursuite de position

Paramètres programmables pour le mode de poursuite de position:

- "Interrupteurs limiteurs actifs"
- "Evaluation d'encodeur"
- "Réglage d'encodeur"
- "Rampe" (uniquement pour présélection de la valeur de position au moyen d'une variable)
- "Signaux interface d'engrenage"
- "Accélération standard" (uniquement pour présélection de la valeur de position au moyen d'une variable)

Le chapitre 3.3.1 décrit la signification et le pré-réglage des paramètres.

Instructions pour la programmation du déplacement en mode de poursuite de position:

Instruction	Signification
"gearn"	Régler facteur d'engrenage dénominateur
"gearz"	Régler facteur d'engrenage numérateur
"goff"	Introduire déplacement de position
"mode"	Régler mode d'exploitation d'axe (mode de poursuite de position)
"stop"	Arrêter déplacement d'axe

Les mouvements sont déterminés en mode de poursuite de position au moyen d'un signal d'impulsion sur la connexion d'encodeur ou à l'aide de la valeur d'une variable.

## Description générale

---

L'instruction "mode" sert à déterminer si la valeur de position est pré-sélectionnée via la connexion d'encodeur ou avec une variable.

- Connexion d'encodeur  
La connexion d'encodeur est sélectionnée au moyen du paramètre "réglage d'encodeur". La fréquence du signal d'impulsion définit l'accélération et la vitesse de l'axe. Si des impulsions ne sont pas disponibles, l'axe est à l'arrêt.
- Variable  
La valeur d'une variable est interprétée comme position absolue pour l'axe. Si cette valeur change l'axe se déplace à la nouvelle position en utilisant la courbe d'accélération sélectionnée (rampe) et la vitesse réglée.



### NOTE

*Le programme-produit OED3 livré comporte un problème-exemple pour le mode de poursuite de position.*

## 1.7 Programmation de l'encodeur

Chaque connexion d'encodeur peut être utilisée pour

- l'introduction de grandeurs de référence (en mode de poursuite de position) ou
- le contrôle de rotation.



**NOTE**

Le programme-produit OED3 livré comporte des problèmes-exemples pour la programmation de l'encodeur.

La figure 1-10 montre la programmation de l'encodeur et son effet sur la commande.

### Evaluation d'encodeur

Lorsqu'un encodeur est utilisé pour l'introduction de grandeurs de référence ou le contrôle de rotation, la résolution de l'encodeur (500 ou 1000 traits) et le facteur d'évaluation interne (simple, double, quadruple) doivent être communiqués à la commande au moyen du paramètre "évaluation d'encodeur".

Il s'applique:

$$\text{Révol. du moteur} = \text{impulsions d'encodeur} \times \frac{\text{facteur d'évaluation}}{\text{résolution d'encodeur}}$$

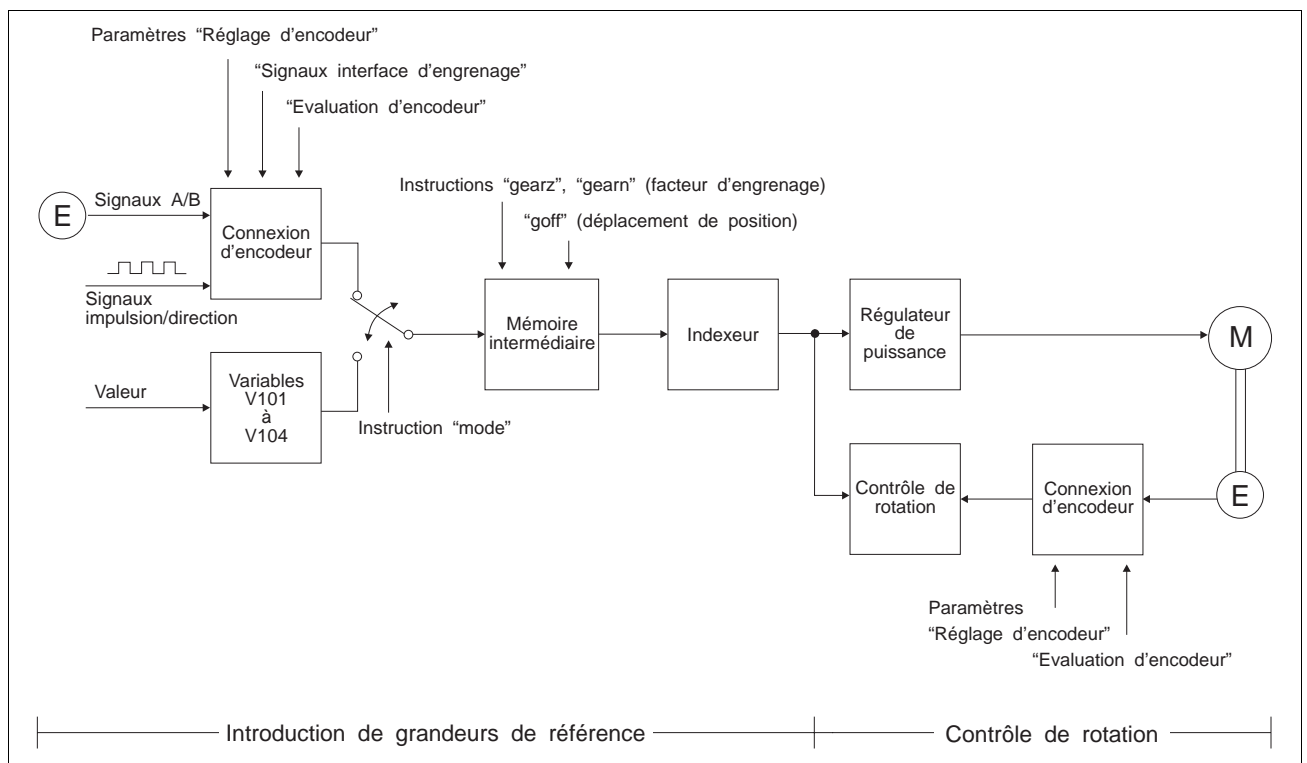


Fig. 1-10 Programmation de l'encodeur

### Introduction de grandeurs de référence (mode de poursuite de position)

En mode de poursuite de position, la grandeur de référence (position) est prédéfinie à l'aide d'une entrée d'encodeur ou d'une variable.

#### Engrenage électronique

La grandeur de référence est multipliée par un facteur d'engrenage, ainsi assurant la réalisation d'un engrenage électronique.

De plus, la grandeur de référence peut être ajustée avec une valeur de position en utilisant l'instruction "goff"(déplacement de position).

Si la grandeur de référence change trop vite, la courbe d'accélération programmée est utilisée pour l'accélération ou la décélération. Une mémoire intermédiaire assure que des impulsions ne soient pas perdues.



#### NOTE

Si l'entraînement est arrêté ou un autre mode d'exploitation d'axe sélectionné, les impulsions introduites (positions) continuent d'être saisies et mémorisées dans la mémoire intermédiaire. Si l'on retourne au mode de poursuite de position, les impulsions sont extraites ce qui entraîne le positionnement relatif de l'entraînement.

Les paramètres et instructions suivants sont importants pour l'introduction de grandeurs de référence:

Paramètre	Signification
"Evaluation d'encodeur"	Ce paramètre sert à régler l'évaluation simple, double ou quadruple des signaux d'encodeur (se reporter au chapitre 3.3.1)
"Réglage d'encodeur"	Ce paramètre sert à régler: 0 Contrôle de rotation n'est pas disponible Connexion d'encodeur 1 pour mode de poursuite de position 1 Contrôle de rotation via connexion d'encodeur 1 Connexion d'encodeur 2 pour mode de poursuite de position 2 Contrôle de rotation via connexion d'encodeur 2 Connexion d'encodeur 1 pour mode de poursuite de position
"Signaux interface d'engrenage"	Ce paramètre sert à régler: 0 Signaux impulsion/direction 1 Signaux A/B

Instruction	Signification
"gearn"	Facteur d'engrenage dénominateur en unités définies par l'utilisateur
"gearz"	Facteur d'engrenage numérateur en unités définies par l'utilisateur
"goff"	Déplacement de position en unités définies par l'utilisateur
"mode"	Régler mode d'exploitation d'axe (mode de poursuite de position)



#### NOTE

Le programme-produit OED3 livré comporte un problème-exemple pour le mode de poursuite de position.

### Contrôle de rotation

Il est possible d'éviter des erreurs de position en activant le contrôle de rotation. Cela signifie qu'un encodeur détecte la position réelle du moteur et la compare à la position exigée. Si la différence excède une certaine limite, le moteur est décélééré.

Les paramètres suivants sont importants pour la programmation du contrôle de rotation:

Paramètre	Signification
"Evaluation d'encodeur"	Ce paramètre sert à régler l'évaluation simple, double ou quadruple des signaux d'encodeur (se reporter au chapitre 3.3.1)
"Réglage d'encodeur"	Ce paramètre sert à régler: 0 Contrôle de rotation n'est pas disponible Connexion d'encodeur 1 pour mode de poursuite de position 1 Contrôle de rotation via connexion d'encodeur 1 Connexion d'encodeur 2 pour mode de poursuite de position 2 Contrôle de rotation via connexion d'encodeur 2 Connexion d'encodeur 1 pour mode de poursuite de position

### Comportement en cas d'erreur de rotation

1. Le moteur se décélère en utilisant la rampe réglée.
2. L'unité de puissance se déconnecte.



#### **ATTENTION**

***Le moteur n'a plus de couple de retenue.***

3. Le message d'erreur "12" apparaît dans l'indicateur d'état 40.
4. La commande se commute à l'état STOP.
5. Toutes les sorties sont commutées à l'état sans courant.
6. Eliminer la cause de l'erreur, p.ex. blocage mécanique, encrassement.
7. Comme la position n'a pas été mémorisée, lancer le programme à nouveau au moyen d'une course de référence.

### 1.8 Interpolation avec unités de positionnement multiaxes

Les unités de positionnement multiaxes de la série 300 (p.ex. WPM-311) permettent de commander simultanément plusieurs axes. Les axes peuvent être commandés soit l'un étant indépendant de l'autre, soit avec une dépendance définie (de manière interpolante). L'interpolation linéaire est possible avec deux ou trois axes.



#### NOTE

*L'interpolation linéaire peut se réaliser avec des valeurs absolues (valeurs de position se réfèrent aux zéros des axes) ou avec des valeurs relatives (valeurs de position se réfèrent à la position actuelle des axes) et sans ou avec état d'attente (le programme est temporairement arrêté jusqu'à ce que l'interpolation soit terminée).*

Les fonctions d'interpolation suivantes sont possibles avec des unités de positionnement multiaxes:

Instruction	Signification
"linpos"	Lancement d'une interpolation linéaire vers une position absolue
"linposf"	Lancement d'une interpolation linéaire vers une position absolue et attendre jusqu'à ce que la position ait été atteinte
"linmove"	Lancement d'une interpolation linéaire vers une position relative
"linmovef"	Lancement d'une interpolation linéaire vers une position relative et attendre jusqu'à ce que la position ait été atteinte
"setipos"	Une position en unités définies par l'utilisateur est prédéfinie pour chaque axe qui doit prendre part à l'interpolation linéaire, p.ex. setipos x1 1000 setipos x2 2000 setipos x3 2000

L'instruction "stopa" sert à arrêter une interpolation linéaire.

Les points suivants doivent être observés pour l'interpolation:

- les axes qui prennent part à l'interpolation doivent être réglés au mode point à point (voir instruction "mode");
- les vitesses et accélérations des axes participants doivent être introduites avant de réaliser l'interpolation (voir instructions "vel" et "acc");
- les positions ainsi que les valeurs de vitesse et d'accélération des axes participants ne peuvent pas être modifiées pendant l'interpolation;
- il n'est pas possible de réaliser plusieurs interpolations simultanément;
- l'interpolation exerce une influence sur l'engrenage électronique (l'engrenage électronique doit éventuellement être réinitialisé après l'interpolation).

### Principe de l'interpolation linéaire

La figure 1-11 montre le principe de l'interpolation linéaire à l'aide d'un exemple:

Un déplacement du point A au point B doit être réalisé avec deux axes (x1 et x2) en utilisant l'interpolation linéaire.

Lors de l'introduction (p.ex. instructions "setipos x1 500" et "setipos x2 200"), les positions exigées des axes sont transmises et l'interpolation est lancée en utilisant l'instruction "linmove".

L'interpolateur linéaire calcule les vitesses et accélérations nécessaires à l'interpolation en utilisant les positions exigées et commande les axes individuels. Il assure que les vitesses et accélérations pré-réglées des axes individuels ne soient pas excédées.

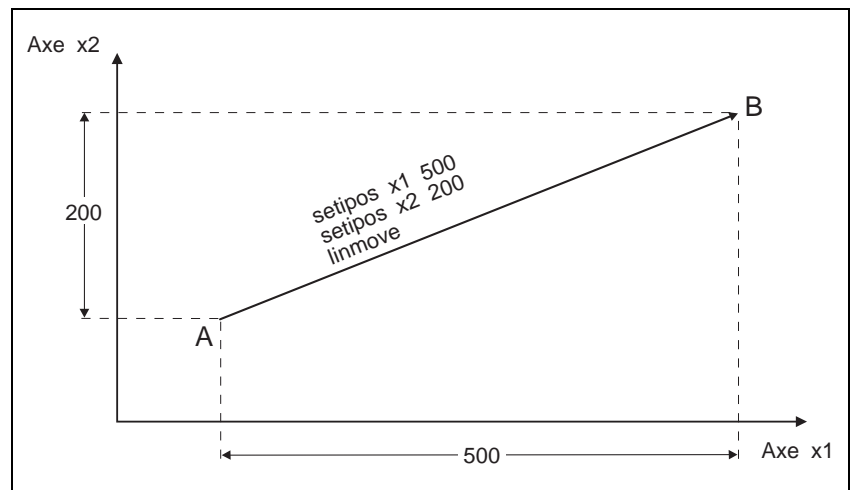


Fig. 1-11 Principe de l'interpolation linéaire

## 1.9 Régulation de position pour WDP3-337 et WDP3-338

S'il s'agit des dispositifs WDP3-337 et WDP3-338, la position réelle de l'axe est détectée via la connexion de résolveur et acheminée au régulateur de position interne (fig. 1-12). La connexion de résolveur est automatiquement réglée à l'évaluation quadruple. Lorsqu'un résolveur avec une résolution de 1024 incréments est utilisé, il en résulte une résolution interne de 4096 unités d'encodeur (incrément) par rotation. La limite d'erreur de poursuite standard est réglée à 4096 unités d'encodeur. Si l'écart de poursuite (différence entre position exigée et position réelle) dépasse la limite d'erreur de poursuite, l'alimentation électrique du moteur est coupée.



### ATTENTION

**Un moteur sans courant n'a aucun couple de retenue. Il peut en résulter l'endommagement de la mécanique. Des charges verticales sur l'axe doivent être protégées contre la chute (moteur avec frein).**

Un régulateur de position PID est intégré dans les commandes WDP3-337 et WDP 3-338. Le réglage des paramètres du régulateur est décrit dans la documentation du dispositif.



### NOTE

Les valeurs pour les paramètres du régulateur peuvent être déterminées au moyen du logiciel de mise en service ONLINE3. Les paramètres déterminés doivent ensuite être introduits dans le programme d'édition de paramètres de l'OED3. Les paramètres ne peuvent pas être modifiés pendant le positionnement.

La connexion d'encodeur peut être utilisée pour l'introduction de grandeurs de référence en mode de poursuite de position, voir chapitre 1.7.

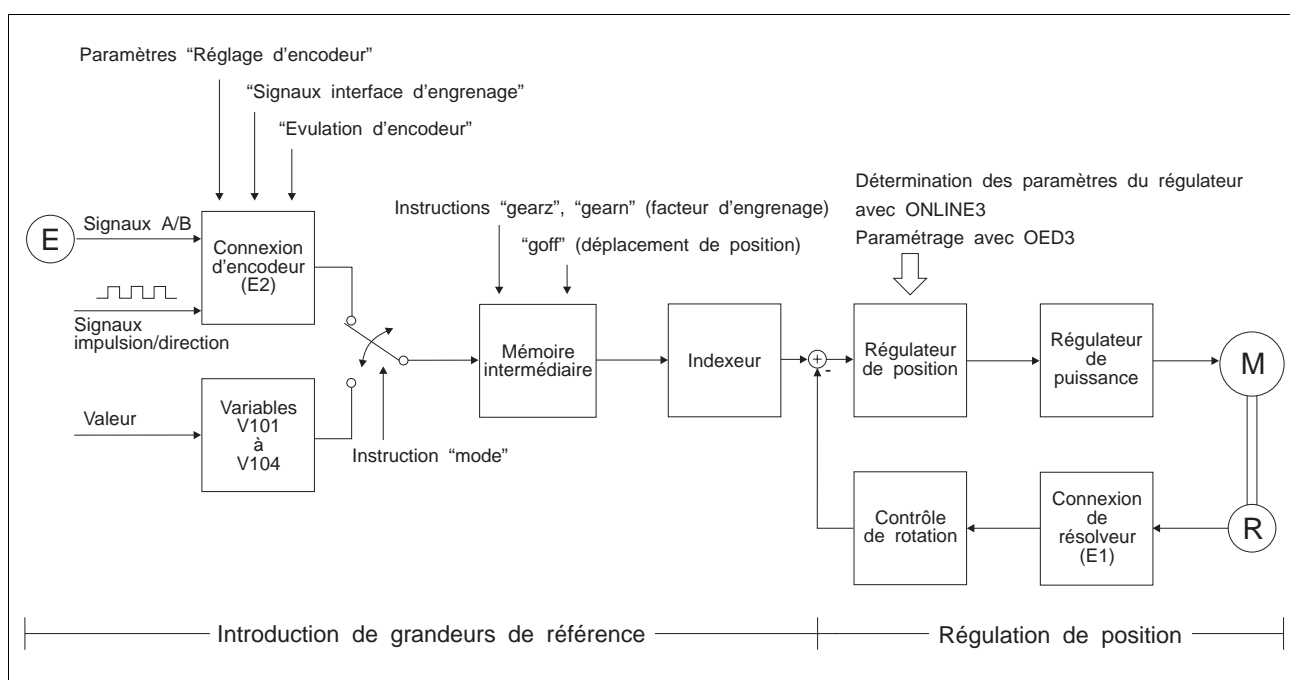


Fig. 1-12 Régulation de position



## 1.10 Programmation d'interface

La commande peut être programmée et maniée à l'aide de l'interface sérielle c1 ou c2, p.ex. en utilisant le terminal de commande FT 2000. Le dialogue d'opérateur peut être programmé indépendamment. L'interface c1 ou c2 est pré-réglée aux valeurs suivantes:

Vitesse baud	9600
Bits utiles	7
Parité	paire
Bits d'arrêt	1
Etablissement d'une liaison	XON/XOFF

Le paramètre "unité de commande" peut être utilisé pour déterminer comment la programmation et le maniement via les interfaces sérielles doivent se réaliser.

### Interface sérielle c1

Le paramètre "unité de commande" signifie pour une commande munie d'une interface sérielle:

Paramètre	Signification
"Unité de commande"	0 Programmation avec OED3 ou dialogue d'opérateur avec programme utilisateur et sortie de plusieurs lignes
	1 Programmation avec OED3 ou dialogue d'opérateur avec programme utilisateur et sortie d'erreurs de deux lignes (p.ex. sur FT 2000)

### Interfaces sérielles c1 et c2

Le paramètre "unité de commande" signifie pour une commande munie de deux interfaces sérielles:

Paramètre	Signification
"Unité de commande"	0 Programmation avec OED3 ou dialogue d'opérateur avec programme utilisateur et sortie de plusieurs lignes via interface c2; Interface c1 sans fonction.
	1 Dialogue d'opérateur avec programme utilisateur et sortie de deux lignes (p.ex. sur FT 2000) via interface c1; Programmation avec OED3 via interface c2.

### Transmission de caractères via l'interface sérielle

Il est possible de transmettre de caractères individuels en utilisant les instructions listées ci-après:

Instruction	Signification
"rec_char"	Recevoir caractères
"rec_var"	Recevoir variables
"snd_str"	Transmettre chaîne de caractères
"snd_var"	Transmettre variables



#### NOTE

La chaîne de caractères qui doit être transmise en utilisant l'instruction "snd\_str", doit être introduite au moyen du programme d'édition de texte OED3, se reporter au chapitre 3.6.

# Description générale

Interprétation des caractères:

Caractères dans le tampon d'entrée/sortie de la commande	Direction de transmission	Caractères via interface	
Caractères ASCII (16#20 - 16#7E)	↔	Caractères ASCII (16#20 - 16#7E)	
Caractères supplémentaires (16#7F - 16#FF)	↔	16#7F - 16#FF	
\$\$	↔	\$ (16#24)	*
\$'	↔	' (16#27)	*
\$L	↔	<LF> (16#0A) Linefeed	*
\$N	↔	(16#0A) Newline	*
\$P	↔	<FF> (16#0C) Formfeed	*
\$R	↔	<CR> (16#0D) Carriage-Return	*
\$T	↔	<TAB> (16#09) Tabulator	*
\$00 - \$1F	←	16#00 - 16#1F	*
\$00 - \$FF	→	16#00 - 16#FF	*

\* Ces caractères sont convertis pendant leur transmission.



**NOTE**

Le programme-produit OED3 livré comporte un problème-exemple pour FT 2000.

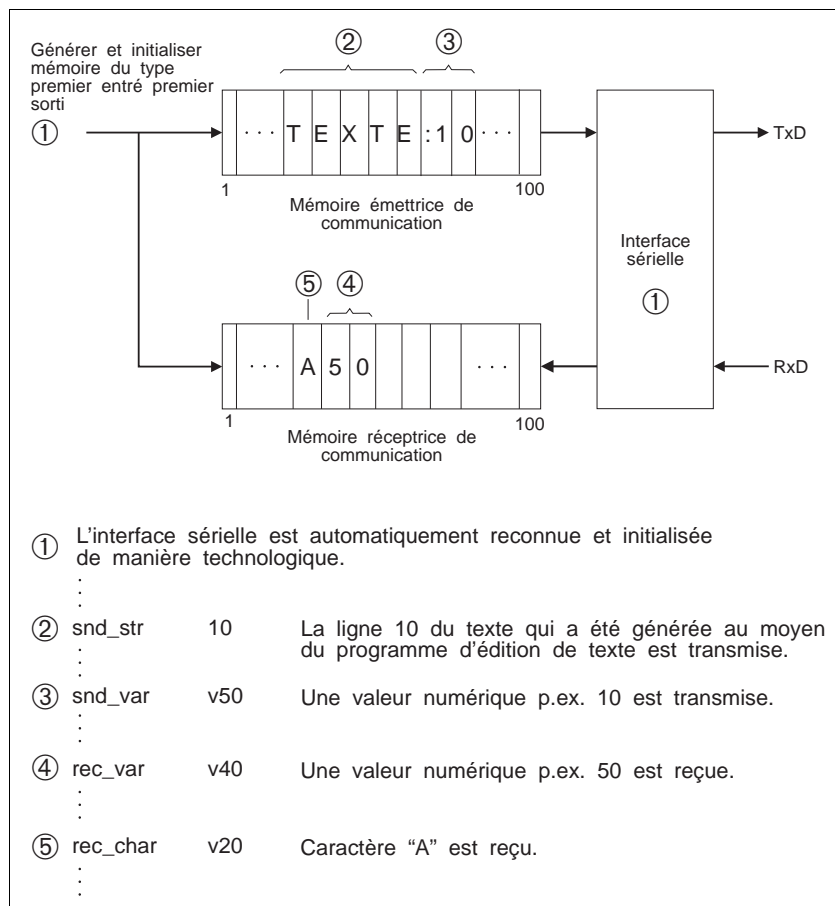


Fig. 1-13 Exemple d'une programmation d'interface

### 1.11 Synchronisation avec la commande prioritaire

Une synchronisation avec la commande prioritaire peut être programmée via une entrée ou sortie facultative en utilisant l'instruction "handshake". La sortie de synchronisation signale qu'une position programmée a été atteinte dans le programme utilisateur. Le programme utilisateur s'arrête à cette position et attend jusqu'à ce que l'entrée de synchronisation soit parcourue par le courant (= 1). Quand l'entrée = 1, la commande continue d'exécuter le programme utilisateur et la sortie de synchronisation est remise à zéro.

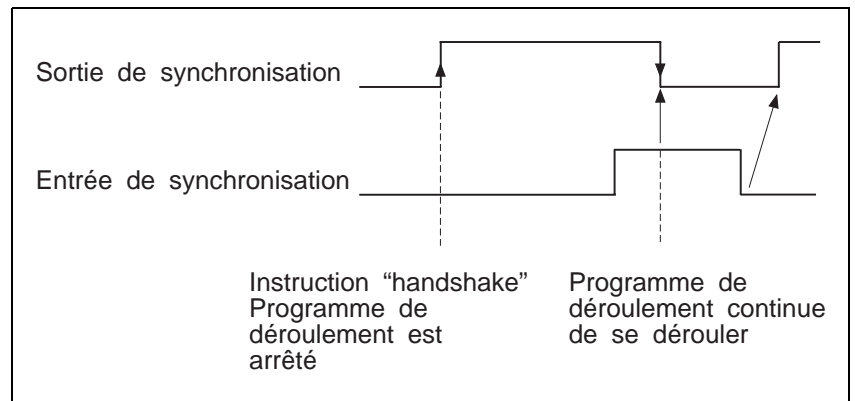


Fig. 1-14 Synchronisation

## 1.12 Exploitation avec carte d'entrée/sortie MP 926

Une carte d'entrée/sortie MP 926 dispose de 16 entrées et 16 sorties qui peuvent être adressées par une unité de positionnement BERGER LAHR de la série 300 qui est équipée de l'interface RS 485 HS.

Le paramètre 35 sert à régler le nombre des cartes d'entrée/sortie raccordées (cinq cartes maxi).

Les indicateurs m32 à m111 sont attribués aux entrées, et les indicateurs m112 à m191 aux sorties (voir fig. 1-15).

Les indicateurs peuvent être lus et écrits dans le programme d'édition de déroulement et API.

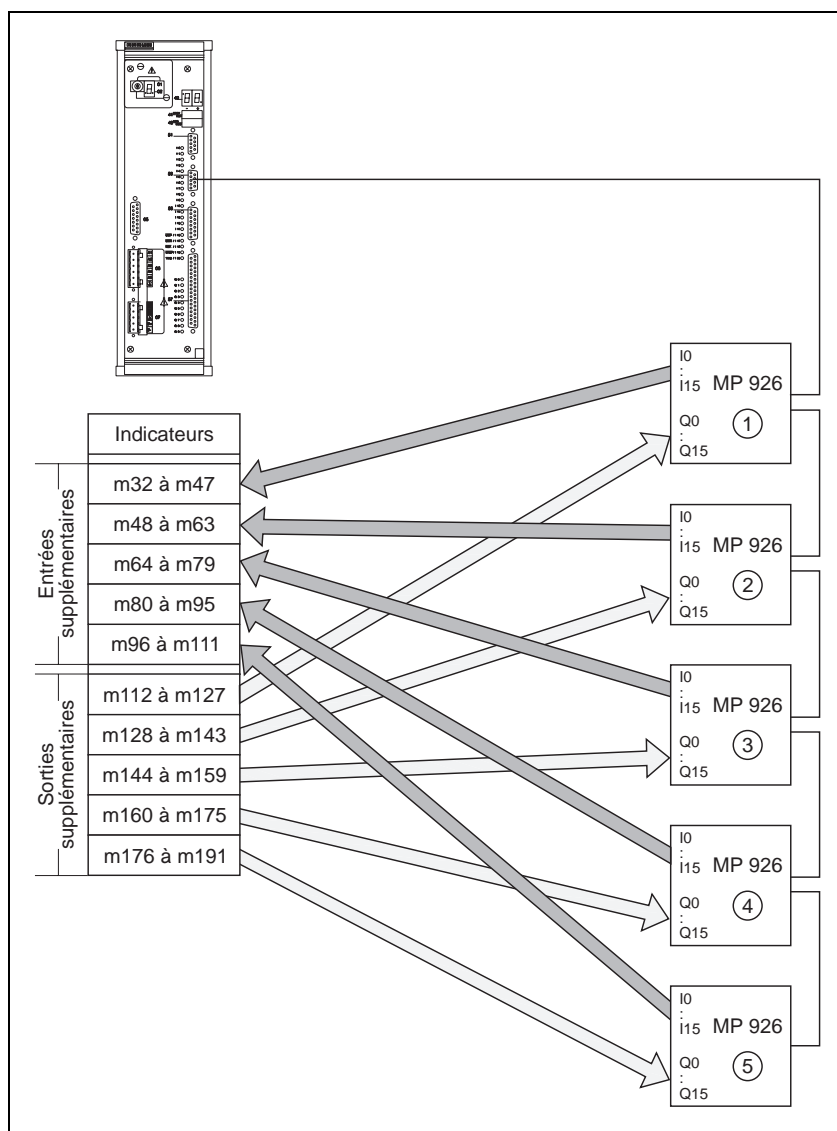


Fig. 1-15 Zones des indicateurs pour entrées/sorties externes

## 2 Installation

### 2.1 Volume de livraison

Contrôler si le volume de livraison est complet.

Le volume de livraison comprend:

Qté	Désignation
1	Dispositif de la série 300 (p.ex. WDP5-318)
1	Disquette 3½" problèmes-exemples OED3 et BTERM
1	Disquette 5¼" problèmes-exemples OED3 et BTERM
1	Documentation du dispositif
1	Documentation OED3
1	Documentation BTERM

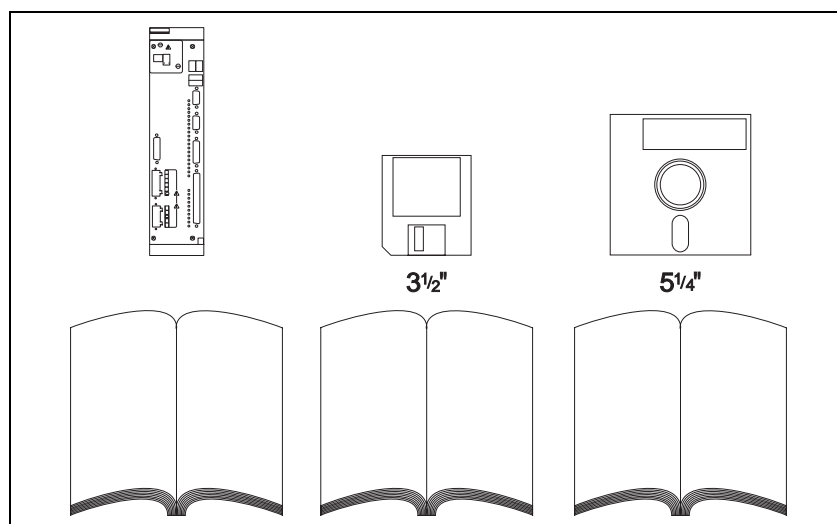


Fig. 2-1 Volume de livraison

### 2.2 Accessoires

Les accessoires listés ci-après (se reporter à la documentation du dispositif, chapitre 6.3) peuvent être livrés sur commande spéciale:

- Câble d'interface prise mâle-prise femelle
- Câble d'interface prise mâle-prise mâle
- Convertisseur d'interface MP 923 (RS 485/RS 232)

## 2.3 Conditions

Conditions pour le maniement de la surface de commande:

- Terminal ou ordinateur personnel avec programme de terminal p.ex. BTERM, se reporter au volume de livraison
- Câble d'interface pour la communication avec les commandes
- évent. convertisseur d'interface RS 485/RS 232 (p.ex. MP 923)
- Commande de la série 300 avec OED3

## 2.4 Schéma de connexion

La communication entre l'ordinateur personnel et la commande se réalise par transmission des données en série en utilisant l'interface c1 ou c2. La figure 2-2 montre le schéma de connexion pour le câblage de la commande avec l'ordinateur personnel.



### NOTE

*S'il s'agit de dispositifs de la série 300 avec interface série, la commande est programmée via l'interface série c1.*

*S'il s'agit de dispositifs équipés de deux interfaces série, la commande est programmée via l'interface série c2. L'interface série c1 peut être utilisée à des autres fins (se reporter au chapitre 1.10).*

*Le câblage des interfaces est décrit dans les chapitres 2 et 6 de la documentation du dispositif.*



### ATTENTION

**En réalisant le câblage on doit considérer si la commande est équipée d'une interface RS 485 (prise femelle) ou d'une interface RS 232 (prise mâle).**

**Si la commande est équipée d'une interface RS 485 et l'ordinateur personnel d'une interface RS 232, un convertisseur d'interface (p.ex. MP 923) doit être utilisé.**

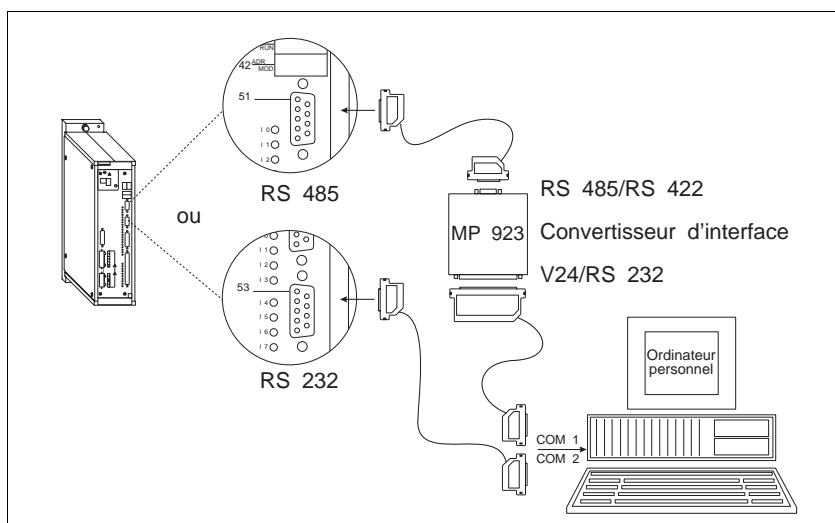


Fig. 2-2 Schéma de connexion p.ex. pour commande WDP5-318

### 3 Programmation

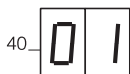
#### 3.1 Lancer OED3

La programmation et le maniement avec la surface de commande OED3 sont maintenant possibles à condition que l'installation ait été réalisée selon le chapitre 2.

1. S'il s'agit de dispositifs avec deux interfaces, raccorder le terminal ou l'ordinateur personnel à l'interface série c2, s'il s'agit de dispositifs avec une interface, raccorder à l'interface série c1.
2. Mettre le terminal ou ordinateur personnel en circuit. Si le maniement se réalise avec un ordinateur personnel, lancer le programme de terminal BTERM.
3. Régler le terminal ou l'ordinateur personnel aux paramètres d'interface suivants:

Vitesse baud	9600
Bits utiles	7
Parité	paire
Bits d'arrêt	1
Etablissement d'une liaison	XON/XOFF

4. Mettre la commande en circuit et attendre jusqu'à ce que "01" soit affiché sur l'indicateur d'état 40.



**NOTE**

Lorsqu'il y a un programme utilisateur dans la commande, celui-ci peut être lancé en amenant la touche de sélection 41 en position + (mode automatique).

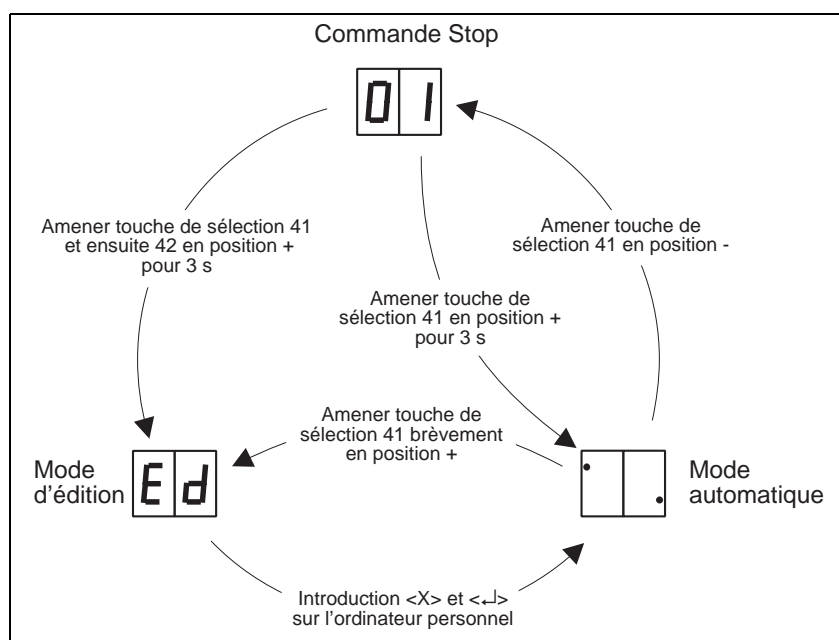
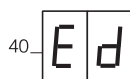


Fig. 3-1 Transitions d'état



5. Amener la touche de sélection 41 en position + (RUN) et, dans l'espace de 3 s, amener la touche de sélection 42 également en position + (MOD) jusqu'à ce que "Ed" apparaisse sur l'indicateur d'état 40. La commande est en mode d'édition. Il est maintenant possible d'introduire ou de traiter un programme utilisateur.



**NOTE**

Lorsqu'un programme utilisateur est exécuté, la touche de sélection 41 doit être amenée en position + (RUN) afin d'activer le mode d'édition.

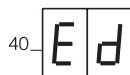
Il apparaît sur l'écran:

```
BERGER LAHR(R)      OED3  PR444.1 V1.01      SIG Positec  
Copyright(C)      BERGER LAHR 1993.  All rights reserved.
```

```
EDIT>
```

*Mode d'édition*

Un programme utilisateur est généré et traité en mode d'édition au moyen des fonctions listées ci-après:



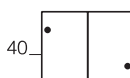
Programme d'édition de paramètres	voir chapitre 3.3
Programme d'édition API	voir chapitre 3.4
Programme d'édition de déroulement	voir chapitre 3.5
Programme d'édition de texte	voir chapitre 3.6

Les fonctions:

Upload	voir chapitre 3.7
Download	voir chapitre 3.8

servent à charger un programme utilisateur depuis la commande dans l'ordinateur ou depuis l'ordinateur dans la commande.

*Mode automatique*



Le mode d'édition peut être abandonné au moyen de <X> et <↵>. Le programme utilisateur (mode automatique) est activé après avoir abandonné le mode d'édition. Deux points apparaissent sur l'indicateur d'état 40.



**NOTE**

Le mode d'exploitation qui était sélectionné lors de la mise hors circuit est à nouveau activé après la mise en circuit de la commande.



## 3.2 Surface de programmation

Ce chapitre décrit les fonctions de commande générales de la surface de programmation OED3.

```

BERGER LAHR(R)      OED3  PR444.1 V1.01      SIG Positec
Copyright(C)      BERGER LAHR 1993.  All rights reserved.

EDIT> H

S Programme d'édition API
T Programme d'édition TXT
A Programme d'édition DER
P Programme d'édition PAR
U Upload
D Download
X FIN

EDIT> _
    
```

Fig. 3-2 Surface de programmation

- La ligne d'en-tête comprend le numéro de programme et de version.
- Le curseur clignotant repère la ligne d'introduction. Toutes les introductions sont validées à l'aide de la touche <↵>.
- La fonction d'aide est appelée en appuyant sur <H>.
- Le programme OED3 est abandonné en appuyant sur <X>.
- Le programme d'édition actuel est abandonné en appuyant sur <ESC>.

Programme d'édition de lignes:

- Effacer une ligne éditée avec <Strg> <Y>.
- Sauter au début de la ligne avec <Strg> <A>.
- Sauter à la fin de la ligne avec <Strg> <F>.
- Un caractère à droite avec <Strg> <D>.



**NOTE**

En introduisant "EEPROM" le programme utilisateur peut être mémorisé dans l'EEPROM de la commande (si cette option est disponible).

- 3.2.1 Programmes d'édition et fonctions** Les programmes d'édition et fonctions listés ci-après sont disponibles:
- Programme d'édition API* Le programme d'édition API sert à générer ou éditer un programme API.
- Programme d'édition de texte* Le programme d'édition de texte sert à sélectionner les textes qui sont visualisés dans le programme de déroulement sur un terminal (p.ex. FT 2000) en utilisant l'instruction "snd\_str".
- Programme d'édition de déroulement* Le programme d'édition de déroulement sert à générer ou éditer le programme de déroulement. Les entrées/sorties sont directement lues et écrites.
- Programme d'édition de paramètres* Le programme d'édition de paramètres sert à régler p.ex. les paramètres de moteur, les paramètres de mouvement et les paramètres de système pour le programme de déroulement.
- Upload* La fonction Upload sert à extraire toutes les données d'utilisateur (p.ex. programme de déroulement, programme API, textes, paramètres) de la commande et à les afficher via l'interface sur l'écran. Les données peuvent être sauvegardées sur l'ordinateur personnel et traitées à l'aide d'un programme d'édition d'ordinateur personnel.  
De plus, il est possible d'introduire des commentaires pour le programme (un commentaire doit commencer par un point-virgule).
- Download* La fonction Download sert à charger les données qui ont été sauvegardées au moyen de la fonction Upload dans la commande et à les soumettre à un contrôle de validité.  
Lorsqu'une erreur se produit, l'opération de chargement dans la commande est interrompue et les données sont rebutées.

### 3.3 Programme d'édition de paramètres

Sélectionner le programme d'édition de paramètres <P> <↓>  
 Appeler la sélection de fonctions <H> <↓>

Les lignes et segments suivants sont indiqués à titre d'exemple.

#### 3.3.1 Afficher les paramètres

Afficher la ligne désirée <L3> <↓>  
 Afficher le segment désiré <L1-17> <↓>

1: Vitesse start/stop	100
2: Vitesse standard	1000
3: Accélération standard	125
4: Vitesse manuelle lente	200
5: Vitesse manuelle rapide	2000
6: Dist. mov. très lent pour mode manuel	10
7: Facteur de cadrage numérateur x1	1000
8: Facteur de cadrage numérateur x2 <sup>1)</sup>	1000
9: Facteur de cadrage numérateur x3 <sup>1)</sup>	1000
10: Facteur de cadrage numérateur x4 <sup>1)</sup>	1000
11: Facteur de cadrage dénominateur x1	1000
12: Facteur de cadrage dénominateur x2 <sup>1)</sup>	1000
13: Facteur de cadrage dénominateur x3 <sup>1)</sup>	1000
14: Facteur de cadrage dénominateur x4 <sup>1)</sup>	1000
15: Rampe x1	0
16: Rampe x2 <sup>1)</sup>	0
17: Rampe x3 <sup>1)</sup>	0
PAR-Edit> _	

Fig. 3-3 Lignes 1 à 17 de la liste de paramètres

18: Rampe x4 <sup>1)</sup>	0
19: Interrupteurs limiteurs actifs x1	3
20: Interrupteurs limiteurs actifs x2 <sup>1)</sup>	3
21: Interrupteurs limiteurs actifs x3 <sup>1)</sup>	3
22: Interrupteurs limiteurs actifs x4 <sup>1)</sup>	3
23: Type de la course de référence x1	0
24: Type de la course de référence x2 <sup>1)</sup>	0
25: Type de la course de référence x3 <sup>1)</sup>	0
26: Type de la course de référence x4 <sup>1)</sup>	0
27: Vitesse depuis interrupteur limiteur	200
28: Dist. de libération interr. limiteur	0
29: Réglage d'encodeur <sup>2)</sup>	0
30: Evaluation d'encodeur E1 <sup>2)</sup>	0
31: Evaluation d'encodeur E2	0
32: Signaux interface d'engrenage	1
33: Unité de commande	0
34: Exécution pas à pas	0
PAR-Edit> _	

Fig. 3-4 Lignes 18 à 34 de la liste de paramètres

1) n'est affiché que pour les unités de positionnement multiaxes  
 2) n'est pas affiché pour les unités de positionnement WDP3-33X

27: Vitesse depuis interrupteur limiteur	200
28: Dist. de libération interr. limiteur	0
29: Réglage d'encodeur <sup>1)</sup>	0
30: Evaluation d'encodeur E1 <sup>1)</sup>	0
31: Evaluation d'encodeur E2	0
32: Signaux interface d'engrenage	1
33: Unité de commande	0
34: Exécution pas à pas	0
35: Modules E/S interbus-S <sup>2)</sup>	0
36: Portion proportionnelle KP <sup>3)</sup>	0
37: Portion intégrale KI <sup>3)</sup>	0
38: Portion différentielle KD <sup>3)</sup>	0
39: Portion de compensation de vitesse KF <sup>3)</sup>	0
40: Portion de compensation d'accél. KA <sup>3)</sup>	0
41: Déplacement variable réglante <sup>3)</sup>	0
42: Temps de palpation TA <sup>3)</sup>	1



PAR-Edit> \_

Fig. 3-5 Lignes 27 à 42 de la liste de paramètres

- 1) n'est pas affiché pour les unités de positionnement WDP3-33X
- 2) n'est affiché que pour les unités de positionnement avec interface RS 485 HS
- 3) n'est affiché que pour les unités de positionnement WDP3-33X

Le tableau suivant comporte les paramètres classés par ordre alphabétique pour le programme de déroulement.

Paramètre	Valeur par défaut	Gamme de réglage/Description
"Accélération standard"	125 Hz/ms	1 à 2000 Hz/ms Si une autre accélération n'est pas présélectionnée dans le programme utilisateur, les axes x1 à x4 sont accélérés ou décélérés en utilisant cette accélération standard.
"Déplacement variable réglante" <sup>1)</sup>	0	±2147483647 Compensation pour éviter une dérive du moteur.
"Distance de libération interrupteur limiteur"	0 pas	0 à 100 pas Après avoir réalisé la course de référence avec succès le moteur est écarté de l'interrupteur limiteur ou de référence atteint en utilisant cette valeur.
"Distance en mouvement très lent pour mode manuel"	10 pas	0 à 100 pas Nombre de pas après changement d'état 0 → 1 des indicateurs m0 et m1.
"Evaluation d'encodeur E1 ou E2" <sup>2)</sup>	0	0 = encodeur 500, évaluation simple 1 = encodeur 500, évaluation double 2 = encodeur 500, évaluation quadruple 3 = encodeur 1000, évaluation simple 4 = encodeur 1000, évaluation double 5 = encodeur 1000, évaluation quadruple

Paramètre	Valeur par défaut	Gamme de réglage/Description
"Exécution pas à pas"	0	<p>0 à 1000            0 = exécution du programme de déroulement sans arrêt            1 à 1000 = exécution pas à pas du programme de déroulement à partir de la ligne indiquée</p> <p>Si la valeur 0 est introduite, le programme de déroulement est exécuté sans arrêt.</p> <p>Le numéro de ligne dans le programme de déroulement est indiqué à partir duquel l'exécution pas à pas doit être réalisée. Une ligne est exécutée avec chaque &lt;↵&gt; à travers l'interface sérielle.</p> <p> <b>NOTE</b>  <i>Les lignes de programme sont sorties, si les indicateurs m9 et m12 sont positionnés.</i></p> <p>Le mode pas à pas est repositionné lors du lancement du programme de déroulement et activé dès que la ligne sélectionnée a été atteinte.</p>
"Facteur de cadrage dénominateur x1 à x4" <sup>3)</sup>	1000	<p>±2147483647</p> <p>Le facteur de cadrage sert à convertir les unités définies par l'utilisateur (p.ex. mm) en unités d'entraînement (pas ou incréments) en mode point à point.</p>
"Facteur de cadrage numérateur x1 à x4" <sup>3)</sup>	1000	
"Interrupteurs limiteurs actifs x1 à x4" <sup>3)</sup>	3	<p>0 = un interrupteur limiteur n'est pas actif            1 = interrupteur limiteur négatif est actif            2 = interrupteur limiteur positif est actif            3 = les deux interrupteurs limiteurs sont actifs</p> <p>Après avoir atteint l'interrupteur limiteur, l'entraînement est arrêté en utilisant la rampe pré-réglée. La commande est ensuite à l'état d'erreur.</p>
"Modules E/S interbus-S" <sup>4)</sup>	0	<p>0 à 5</p> <p>Nombre de cartes d'entrée/sortie MP 926 raccordées sur l'interface RS 485 HS (c2)            (0 = une carte n'est pas raccordée)</p> <p> <b>NOTE</b>  <i>La valeur doit être conforme aux cartes d'entrée/sortie raccordées.</i></p>
"Portion de compensation d'accélération KA" <sup>1)</sup>	0	<p>±16383</p> <p>La portion de compensation d'accélération sert à compenser l'écart de poursuite qui dépend de l'accélération.</p>
"Portion de compensation de vitesse KF" <sup>1)</sup>	0	<p>±16383</p> <p>La portion de compensation de vitesse sert à compenser l'écart de poursuite qui dépend de la vitesse.</p>
"Portion différentielle KD" <sup>1)</sup>	0	<p>±16383</p> <p>La portion différentielle définit la réaction aux modifications des déviations de réglage.</p>

## Programmation

Paramètre	Valeur par défaut	Gamme de réglage/Description
"Portion intégrale KI" <sup>1)</sup>	0	0 à 16383 La portion intégrale compense une déviation de réglage permanente.
"Portion proportionnelle KP" <sup>1)</sup>	0	0 à 16383 La portion proportionnelle définit la rigidité de l'entraînement.
"Rampe x1 à x4" <sup>3)</sup>	1	0 = rampe linéaire 1 = rampe exponentielle 2 = rampe sin <sup>2</sup> Courbe utilisée pour l'accélération et la décélération.
"Réglage d'encodeur" <sup>2)</sup>	0	0 = aucune contrôle de rotation connexion d'encodeur 1 pour mode de poursuite de position 1 = contrôle de rotation via connexion d'encodeur 1 connexion d'encodeur 2 pour mode de poursuite de position 2 = contrôle de rotation via connexion d'encodeur 2 connexion d'encodeur 1 pour mode de poursuite de position
"Signaux interface d'engrenage"	1	0 = Signaux impulsion/direction 1 = Signaux A/B Signaux d'entrée sur la connexion d'encodeur pour le mode de poursuite de position
"Temps de palpation TA" <sup>1)</sup>	1	1 = 1 ms, 2 = 2 ms, 3 = 4 ms, 4 = 8 ms Intervalle pendant lequel la position exigée est comparée à la position réelle et une nouvelle variable réglante est calculée par le régulateur de position PID.
"Type de la course de référence x1 à x4" <sup>3)</sup>	0	0 = vers interrupteur limiteur négatif 1 = vers interrupteur limiteur positif 2 = vers interrupteur de référence, rotation à gauche 3 = vers interrupteur de référence, rotation à droite Définit quel axe doit s'approcher de quel interrupteur limiteur.
"Unité de commande"	0	0 = Terminal ASCII de 24 lignes et 80 caractères/ligne 1 = Terminal de deux lignes (p.ex. FT 2000)
"Vitesse depuis interrupteur limiteur"	200 Hz	1 à 10000 Hz Vitesse utilisée pour abandonner l'interrupteur limiteur atteint.
"Vitesse manuelle lente"	200 Hz	1 à 10000 Hz Fréquence utilisée pour le positionnement en mode manuel.
"Vitesse manuelle rapide"	2000 Hz	1 à 10000 HZ Fréquence utilisée pour le positionnement en mode manuel rapide.
"Vitesse standard"	10000 Hz	1 à 100000 Hz Si une autre vitesse n'est pas spécifiée dans le programme utilisateur, tous les axes utilisés x1 à x4 sont déplacés à cette fréquence maximale.
"Vitesse start/stop"	100 Hz	1 à 10000 Hz Vitesse utilisée pour démarrer ou arrêter l'axe.

1) n'est affiché que pour les unités de positionnement WDP3-33X

2) n'est pas affiché pour les unités de positionnement WDP3-33X

3) "x2 à x4" n'est affiché que pour les unités de positionnement multiaxes

4) n'est affiché que pour les unités de positionnement avec interface RS 485 HS

**3.3.2 Editer les paramètres**

Appeler la ligne de paramètre <E1> <↓>  
 Parcourir la liste de paramètres en feuilletant <↓>  
 Terminer l'édition <ESC>



**NOTE**

*Les paramètres modifiés restent en mémoire après la mise hors circuit.*

```

PAR-EDIT> E1
  1: Vitesse start/stop           100
  2: Vitesse standard             10000
  3: Accélération standard        125
  4: Vitesse manuelle lente       200
  5: Vitesse manuelle rapide      2000
  6: Dist. mouv. très lent pour mode manuel 10
    
```

*Fig. 3-6 Editer la ligne 6 des paramètres*

**Exemple:**

Modifier la distance en mouvement très lent à "20":

PAR-Edit>E6 <↓> <20> <↓> <ESC>

## 3.4 Programme d'édition API

Sélectionner le programme d'édition API	<S>	<↵>
Appeler la sélection de fonctions	<H>	<↵>
Insérer	voir chapitre 3.4.3	
Effacer	voir chapitre 3.4.4	
Afficher	voir chapitre 3.4.2	
Editer	voir chapitre 3.4.5	
Liste d'instructions	voir chapitre 3.4.1	

Les lignes et segments suivants sont indiqués à titre d'exemple.

```

BERGER LAHR(R)      OED3  PR444.1 V1.01      SIG Positec
Copyright(C)      BERGER LAHR 1993.  All rights reserved.

EDIT> S
API-Edit> H

I Insérer
D Effacer
L Afficher
E Editer
B Liste d'instructions

API-Edit> _
    
```

Fig. 3-7 Programme d'édition API

### 3.4.1 Liste d'instructions

Appeler la liste d'instructions	<B>	<↵>
---------------------------------	-----	-----

```

API-Edit> B

nop          add      <kv>      and      <iqm>
andn  <iqm>  ld      <kviqmb>  ldn      <iqmb>
or      <viqm>  orn     <viqm>      r        <qm>
s       <qm>    st      <viqmx>  stn      <iqm>
sub    <kv>    eq      <kviqm>  gt       <kv>
lt     <kv>    jmp     <lk>    jmpc    <lk>
jmpn   <lk>    mul    <kv>    div     <kv>
label  <l>    stimer <t>      end

API-Edit> _
    
```

Fig. 3-8 Liste d'instructions du programme d'édition API

Une instruction dans la liste d'instructions est toujours décrite de la manière suivante:

	Opérateur	<opérandes possibles>
	ld	<k>
p.ex.	ld	100

Les tableaux suivants décrivent les opérateurs et opérandes possibles.



Opérandes possibles:

Opérande	Signification		Gamme de valeurs					
<b>b</b>	<b>Valeur booléenne</b>		0 ou 1					
<b>i</b>	<b>Entrées</b>		0 ou 1					
	i0 à i15	pour WDP3-33X, WDP5-318, WP-311						
	i0 à i20	pour WPM-311						
<b>k</b>	<b>Constantes</b> (valeur)		$\pm 2 \times 10^9$					
<b>l</b>	<b>Label</b> l1 à l50							
<b>m</b>	<b>Indicateurs</b>		0 ou 1					
	m0	Course manuelle à droite						
	m1	Course manuelle à gauche						
	m2	Course rapide						
	m3	Sélection multiaxes bit 0				axe	m4	m3
	m4	Sélection multiaxes bit 1				1	0	0
						2	0	1
						3	1	0
						4	1	1
	m5	Activer teach-in						
	m6	Prendre position en charge						
	m7	Ne pas prendre position en charge						
	m8	Réservé						
	m9	Sortie de test: 0 = arrêt, 1 = marche						
	m10	Activer manuel: 0 = non activé, 1 = activé						
	m11	Course manuelle activée? 0 = non, 1 = oui						
	m12	Sortie de test: 0 = numéro de ligne, 1 = ligne complète						
	m13 à m20	Réservé						
	m21 à m1000	Librement disponibles; pour commande avec interface RS 485 HS ou interbus-S:						
	m32 à m111	pour les entrées étendues						
	m112 à m191	pour les sorties étendues						
	m1001	Axe x1: 0 = axe à l'arrêt, 1 = axe se déplace						
	m1002	Axe x2: 0 = axe à l'arrêt, 1 = axe se déplace						
m1003	Axe x3: 0 = axe à l'arrêt, 1 = axe se déplace							
m1004	Axe x4: 0 = axe à l'arrêt, 1 = axe se déplace							
m1005 à m1023	Réservé							
<b>q</b>	<b>Sorties</b> q0 à q9		0 ou 1					
<b>t</b>	<b>Timer</b> t1 à t5 (résolution 100 ms)		0 à $2 \times 10^9$					
<b>v</b>	<b>Variables</b>		$\pm 2 \times 10^9$					
	v0 à v100	Librement disponibles						
	v101	Variable de poursuite de position axe x1						
	v102	Variable de poursuite de position axe x2						
	v103	Variable de poursuite de position axe x3						
	v104	Variable de poursuite de position axe x4						
	v105 à v114	Réservé						
<b>x</b>	<b>Axe</b>							
	x1	pour WDP3-33X, WDP5-318, WP-311						
	x1 à x4	pour WPM-311						

Opérateurs possibles pour instructions API:

Opérateur	Signification
"add"	Opération arithmétique addition
"and"	Opération logique: opération ET
"andn"	Opération logique; opération ET inversée
"div"	Opération arithmétique division
"end"	Terminer programme API
"eq"	Opération de comparaison =
"gt"	Opération de comparaison >
"jmp"	Saut à un repère
"jmpc"	Saut conditionnel, si ROL = VRAI (1)
"jmpn"	Saut conditionnel, si ROL = FAUX (0)
"label"	Repère de saut
"ld"	Opération de transfert
"ldn"	Opération de transfert
"lt"	Opération de comparaison <
"mul"	Opération arithmétique multiplication
"nop"	Terminer programme API
"or"	Opération logique: opération OU
"orn"	Opération logique: opération OU inversée
"r"	Remettre à un opérande booléen
"s"	Mettre à un opérande booléen
"st"	Opération de transfert
"stimer"	Timer
"stn"	Opération de transfert
"sub"	Opération arithmétique soustraction

### Résultat d'opération logique ROL

Le résultat d'opération logique ROL est une mémoire intermédiaire (accumulateur) de la commande qui est utilisée pour des opérations arithmétiques, opérations logiques et la transmission de données.

Exemple	Signification
ld 15 add 10 st v10	Charger valeur 15 dans ROL Additionner la valeur 10 au ROL. Mémoriser le résultat en tant que variable v10.

### Description des instructions API

#### ***add***

---

Syntaxe:        *add*                <kv>

L'instruction "add" permet de réaliser l'addition avec un opérande et le résultat d'opération logique ROL.

*Exemple:*

Opérateur	Opérande	Signification
<i>ld</i>	<i>100</i>	Charger valeur 100 dans ROL.
<i>add</i>	<i>300</i>	Additionner la valeur 300 au contenu du ROL (valeur 100).
<i>st</i>	<i>v5</i>	Mémoriser le résultat en tant que variable <i>v5</i> .
:		

#### ***and***

---

Syntaxe:        *and*                <iqm>  
                   *andn*             <iqm>

L'instruction "and" permet de réaliser une opération logique avec un opérande booléen et le ROL.

Une négation booléenne de l'opérande peut être réalisée avant l'opération logique en ajoutant le suffixe "n". Le suffixe "n" doit être ajouté de la manière suivante, p.ex. "andn".

*Exemple 1:*

Opérateur	Opérande	Signification
<i>ld</i>	<i>i1</i>	Charger entrée <i>i1</i> dans ROL.
<i>and</i>	<i>i2</i>	Opération ET avec entrée <i>i1</i> et <i>i2</i> . ROL = <i>i1</i> and <i>i2</i>
<i>st</i>	<i>m15</i>	Mémoriser contenu du ROL en tant qu'indicateur <i>m15</i> .
:		

*Exemple 2:*

Opérateur	Opérande	Signification
<i>ld</i>	<i>i1</i>	Charger entrée <i>i1</i> dans ROL.
<i>andn</i>	<i>i3</i>	Opération ET inversée avec entrée <i>i3</i> et ROL. ROL = ROL andn <i>i3</i>
<i>st</i>	<i>m11</i>	Mémoriser contenu du ROL en tant qu'indicateur <i>m11</i> .
:		

### ***div***

---

Syntaxe:        `div`            `<kv>`

L'instruction "div" permet de réaliser la division avec un opérande et le ROL.

*Exemple:*

Opérateur	Opérande	Signification
<code>ld</code>	<code>100</code>	Charger valeur 100 dans ROL.
<code>div</code>	<code>20</code>	Diviser le contenu du ROL (valeur 100) par la valeur 20.
<code>st</code>	<code>v4</code>	Mémoriser le résultat en tant que variable v4.
:		

### ***end***

---

Syntaxe:        `end`

L'instruction "end" sert à terminer un cycle API. L'image de processus est mise à jour et un nouveau cycle commence à partir de la ligne 1.

*Exemple:*

Opérateur	Opérande	Signification
<code>ld</code>	<code>100</code>	Charger valeur 100 dans ROL.
<code>add</code>	<code>300</code>	Additionner la valeur 300 au contenu du ROL (valeur 100).
<code>st</code>	<code>v5</code>	Mémoriser le résultat en tant que variable v5.
<code>end</code>		Fin de cycle

## **eq**

---

Syntaxe:        eq                    <kvimm>

L'instruction "eq" sert à réaliser une comparaison (=) avec un opérande et le ROL. Après la comparaison, le ROL est toujours:

0 ou FAUX, si la comparaison est fausse  
1 ou VRAI, si la comparaison est vraie.

*Exemple:*

Opérateur	Opérande	Signification
:		
ld	v100	Charger variable v100 dans ROL.
eq	100	Réaliser comparaison. Est-ce que ROL = 100? Si la comparaison est vraie, ROL = 1, autrement ROL = 0.
:		

## **gt**

---

Syntaxe:        gt                    <kv>

L'instruction "gt" sert à réaliser une comparaison (>) avec un opérande et le ROL. Après la comparaison, le ROL est toujours:

0 ou FAUX, si la comparaison est fausse ou  
1 ou VRAI, si la comparaison est vraie.

*Exemple:*

Opérateur	Opérande	Signification
:		
ld	v100	Charger variable v100 dans ROL.
gt	1500	Comparer ROL > 1500. Si la comparaison est vraie, ROL = 1, autrement ROL = 0.
:		

### *jmp*

---

Syntaxe:        jmp            <lk>  
                 jmpc         <lk>  
                 jmpn         <lk>

L'instruction "jmp" sert à réaliser des sauts conditionnels et inconditionnels. Les suffixes "c" et "n" permettent des sauts conditionnels. "k" définit un saut relatif de k lignes.

Des instructions de saut avec suffixe "c" ne sont exécutées que si la valeur dans le ROL est inégale à 0.

Des instructions de saut avec suffixe "n" ne sont exécutées que si la valeur dans le ROL est égale à 0.

#### *Exemple 1:*

Opérateur	Opérande	Signification
:		
label	l1	Attribuer repère de saut l1.
ld	v100	Charger variable v100 dans ROL.
add	100	Additionner valeur 100 au ROL.
st	v100	Mémoriser résultat en tant que variable v100.
jmp	l1	Sauter au repère de saut l1.
:		

#### *Exemple 2:*

Opérateur	Opérande	Signification
:		
ld	i12	Charger entrée i12 dans ROL.
jmpc	l10	Si i12 ≠ 0, le saut au repère de saut l10 est réalisé.
:		
label	l10	Attribuer repère de saut l10.

#### *Exemple 3:*

Opérateur	Opérande	Signification
:		
ld	i10	Charger entrée i10 dans ROL.
jmpn	-1	Si i10 = 0, un retour d'une ligne est réalisé.

### ***label***

---

Syntaxe:        label        <l>

L'instruction "label" sert à attribuer un repère pour des sauts et appels de sous-programme.

*Exemple:*

Opérateur	Opérande	Signification
:		
label	l1	Attribuer repère de saut l1.
ld	v100	Charger variable v100 dans ROL.
add	100	Additionner valeur 100 au ROL.
st	v100	Mémoriser résultat en tant que variable v100.
jmp	l1	Sauter au repère de saut l1.
:		

### ***ld***

---

Syntaxe:        ld            <kvixmxb>  
                   ldn           <iqmb>

L'instruction "ld" permet de charger un opérande dans le ROL. Grâce au suffixe "n", une négation booléenne de l'opérande peut être réalisée.

*Exemple 1:*

Opérateur	Opérande	Signification
:		
ld	i1	Charger entrée i1 dans ROL.
st	m15	Mémoriser contenu du ROL (entrée i1) en tant qu'indicateur m15.
:		

*Exemple 2:*

Opérateur	Opérande	Signification
:		
ldn	i1	Charger entrée inversée i1 dans ROL.
ld	i2	Charger entrée i2 dans ROL.
st	m16	Mémoriser ROL en tant qu'indicateur m16.
:		

### ***lt***

---

Syntaxe:        `lt`                `<kv>`

L'instruction "lt" permet de réaliser une comparaison (<) avec un opérande et le ROL. Après la comparaison, le ROL est toujours:

0 ou FAUX, si la comparaison est fautive ou  
1 ou VRAI, si la comparaison est vraie.

*Exemple:*

Opérateur	Opérande	Signification
:		
<code>ld</code>	<code>v110</code>	Charger variable v110 dans ROL.
<code>lt</code>	<code>1800</code>	Comparer ROL < 1800. Si la comparaison est vraie, ROL = 1, autrement ROL = 0.
:		

### ***mul***

---

Syntaxe:        `mul`                `<kv>`

L'instruction "mul" permet de faire la multiplication avec un opérande et le ROL.

*Exemple:*

Opérateur	Opérande	Signification
<code>ld</code>	<code>100</code>	Charger valeur 100 dans ROL.
<code>mul</code>	<code>20</code>	Multiplier le ROL (valeur 100) par la valeur 20.
<code>st</code>	<code>v3</code>	Mémoriser résultat en tant que variable v3.
:		



## ***nop***

---

Syntaxe:        `nop`

L'instruction "nop" termine le programme API.

## ***or***

---

Syntaxe:        `or`                `<viqm>`  
                  `orn`               `<viqm>`

L'instruction "or" permet de réaliser une opération logique avec un opérande booléen et le ROL.

Une négation booléenne de l'opérande peut être réalisée avant l'opération logique en ajoutant le suffixe "n". Le suffixe "n" doit être ajouté à l'instruction, p.ex. "orn".

### *Exemple 1:*

Opérateur	Opérande	Signification
<code>ld</code>	<code>i1</code>	Charger entrée i1 dans ROL.
<code>or</code>	<code>i2</code>	Opération OU avec entrée i1 et i2. ROL = i1 or i2
<code>st</code>	<code>m17</code>	Mémoriser contenu du ROL en tant qu'indicateur m17.

### *Exemple 2:*

Opérateur	Opérande	Signification
<code>ld</code>	<code>i1</code>	Charger entrée i1 dans ROL.
<code>and</code>	<code>i2</code>	Opération ET avec entrée i1 et i2. ROL = i1 and i2.
<code>orn</code>	<code>i3</code>	Opération OU inversée avec ROL et entrée i3. ROL = ROL orn i3
<code>st</code>	<code>m11</code>	Mémoriser ROL en tant qu'indicateur m11.

### ***r***

---

Syntaxe:        *r*                    <qm>

L'instruction "r" sert à remettre une sortie ou un indicateur à zéro en fonction du résultat d'opération logique.

*Exemple:*

Opérateur	Opérande	Signification
ld	i1	Charger entrée i1 dans ROL.
and	i2	Opération ET avec entrée i1 et i2. ROL = i1 and i2.
st	m11	Mémoriser contenu du ROL en tant qu'indicateur m11.
r	q5	Sortie q5 est remise à 0, si ROL = 1.

### ***s***

---

Syntaxe:        *s*                    <qm>

L'instruction "s" sert à mettre une sortie ou un indicateur à 1 en fonction du résultat d'opération logique.

*Exemple:*

Opérateur	Opérande	Signification
ld	i1	Charger entrée i1 dans ROL.
and	i2	Opération ET avec entrée i1 et i2. ROL = i1 and i2.
st	m13	Mémoriser contenu du ROL en tant qu'indicateur m13.
s	q5	Sortie q5 est mise à 1, si ROL = 1.

### **st**

---

Syntaxe:        st                <viqmx>  
                   stn             <iqm>

L'instruction "st" permet de mémoriser le ROL dans un opérande. Une négation booléenne de l'opérande peut être réalisée en ajoutant le suffixe "n".

*Exemple 1:*

Opérateur	Opérande	Signification
:		
ld	i1	Charger entrée i1 dans ROL.
st	m13	Mémoriser contenu du ROL (entrée i1) en tant qu'indicateur m13.
:		

*Exemple 2:*

Opérateur	Opérande	Signification
:		
ld	i1	Charger entrée i1 dans ROL.
stn	m14	Mémoriser contenu du ROL (négation de l'entrée i1) en tant qu'indicateur m14.
:		

### **stimer**

---

Syntaxe:        stimer        <t>

Des timers sont des variables spécifiques dont la valeur change en fonction du temps. Il existe cinq timers t1 à t5. L'instruction "stimer" sert à charger une valeur de temps dans un timer et à l'activer. La résolution de temps est de 100 ms (temps = valeur de temps x 100 ms). Un timer en cours peut être arrêté au moyen de la valeur 0. Une nouvelle valeur de temps peut être mise à tout moment.

*Exemple:*

Opérateur	Opérande	Signification
:		
ld	10	Charger temps, ici 1 s.
st	t1	Mémoriser valeur de temps dans timer 1.
stimer	t1	Activer timer 1.
:		
ld	t1	
eq	0	Si timer 1 = 0,
st	q1	mettre sortie 1.

### **sub**

---

Syntaxe:       sub            <kv>

L'instruction "sub" sert à réaliser une soustraction avec un opérande et le ROL.

*Exemple:*

Opérateur	Opérande	Signification
ld	100	Charger valeur 100 dans ROL.
sub	30	Soustraire la valeur 30 du contenu du ROL (valeur 100).
st	v6	Mémoriser résultat en tant que variable v6.
:		

**3.4.2 Afficher le programme API**

Afficher le programme API  
Afficher le segment désiré

<L> <↓>  
<L1-17> <↓>

```

API-Edit> l1-17

1:  ld      i1
2:  and     i2
3:  st      m5
4:  s       q5
5:  ldn     i1
6:  orn     i2
7:  r       q5
8:  ld      v100
9:  gt      1500
10: s       q6
11: ld      17
12: ld      true
13: st      q7
14: jmp     l1
15: ld      false
16: st      q7
17: end

API-Edit> _
    
```

*Fig. 3-9 Exemple d'un programme API*

**3.4.3 Insérer une ligne**

Insérer une ligne, p.ex. à partir de la ligne 5  
Insérer et terminer l'édition

<I5> <↓>  
<ESC>

Exemple:

```

API-Edit>I5 <↓>
API-Edit-I0005>LD V15 <↓>
API-Edit-I0006>ADD 100 <↓>
API-Edit-I0007> <ESC>
    
```

### 3.4.4 Effacer ligne(s)

Effacer ligne, p.ex. ligne 5

<D5> <↵>

Effacer lignes, p.ex. lignes 1 à 5

<D1-5> <↵>



#### NOTE

*L'opération d'effacement est immédiatement exécutée et le programme API complet mis à jour.*

Exemple:

```
API-Edit>D5 <↵>
```



#### NOTE

*Après avoir introduit <↵> OK est affiché.*

### 3.4.5 Editer une ligne

Appeler la ligne désirée, p.ex. ligne 11

<E11> <↵>

Terminer l'édition

<ESC>

Exemple:

```
API-Edit>E11 <↵>
API-Edit-0011>LD V100 <↵>
API-Edit-0012>LD V110 <↵>
API-Edit-0013>GT 1500 <↵>
API-Edit-0014> <ESC>
```

### 3.5 Programme d'édition de déroulement

Sélectionner le programme d'édition de déroulement <A> <↓>  
 Appeler la sélection de fonctions <H> <↓>

Insérer	voir chapitre 3.5.3
Effacer	voir chapitre 3.5.4
Afficher	voir chapitre 3.5.2
Editer	voir chapitre 3.5.5
Liste d'instructions	voir chapitre 3.5.1

Les lignes et segments suivants sont indiqués à titre d'exemple.

```

BERGER LAHR(R)      OED3  PR444.1 V1.01      SIG Positec
Copyright(C)      BERGER LAHR 1993.  All rights reserved.

EDIT> A
DER-Edit> H

I Insérer
D Effacer
L Afficher
E Editer
B Liste d'instructions

DER-Edit> _
    
```

Fig. 3-10 Programme d'édition de déroulement

#### 3.5.1 Liste d'instructions

Appeler la liste d'instructions <B> <↓>

```

DER-Edit> B

nop          acc      <x><kv>   add      <kv>
and      <iqm>   andn   <iqm>   div      <kv>
eq      <kviqm>   gearn  <x><kv>   gearz   <x><kv>
goff    <x><kv>   gt      <kv>   handshake <i><q>
jmp     <lk>     jmpc   <lk>   jmpn    <lk>
ld      <kviqmx> lt      <kv>   move    <x><kv>
movef   <x><kv>   mul    <kv>   or      <vism>
orn     <vism>   pos    <x><kv>   posf    <x><kv>
mode    <x><k>   r      <qm>   rec_var <v>
rec_char<v>   ref    <x>   reff    <x>
s       <qm>     snd_str <kv>   snd_var <v>
st      <vismx> stop   <x>   sub     <kv>
vel     <x><kv>   wait  <kv>   cal     <l>
label  <l>     ret    <kv>   end

DER-Edit> _
    
```

Fig. 3-11 Liste d'instructions

## Programmation

Une instruction dans la liste d'instructions est toujours décrite de la manière suivante:

	Opérateur	<opérandes possibles>
	ld	<k>
p.ex.	ld	100

Les tableaux suivants décrivent les opérateurs et opérandes possibles.

Opérandes possibles:

Opérande	Signification				Gamme de valeurs	
<b>b</b>	<b>Valeur booléenne</b>				0 ou 1	
<b>i</b>	<b>Entrées</b>				0 ou 1	
	i0 à i15	pour WDP3-33X, WDP5-318, WP-311				
	i0 à i20	pour WPM-311				
<b>k</b>	<b>Constantes</b> (valeur)				$\pm 2 \times 10^9$	
<b>l</b>	<b>Label</b> l1 à l50					
<b>m</b>	<b>Indicateurs</b>				0 ou 1	
	m0	Course manuelle à droite				
	m1	Course manuelle à gauche				
	m2	Course rapide				
	m3	Sélection multiaxes bit 0	axe	m4		m3
	m4	Sélection multiaxes bit 1	1	0		0
			2	0		1
			3	1		0
			4	1		1
	m5	Activer teach-in				
	m6	Prendre position en charge				
	m7	Ne pas prendre position en charge				
	m8	Réservé				
	m9	Sortie de test: 0 = arrêt, 1 = marche				
	m10	Activer manuel: 0 = non activé, 1 = activé				
	m11	Course manuelle activée? 0 = non, 1 = oui				
	m12	Sortie de test: 0 = numéro de ligne, 1 = ligne complète				
	m13 à m20	Réservé				
	m21 à m1000 m32 à m111 m112 à m191	Librement disponibles; pour commande avec interface RS 485 HS ou interbus-S: pour les entrées étendues pour les sorties étendues				
	m1001	Axe x1: 0 = axe à l'arrêt, 1 = axe se déplace				
m1002	Axe x2: 0 = axe à l'arrêt, 1 = axe se déplace					
m1003	Axe x3: 0 = axe à l'arrêt, 1 = axe se déplace					
m1004	Axe x4: 0 = axe à l'arrêt, 1 = axe se déplace					
m1005 à m1023	Réservé					
<b>q</b>	<b>Sorties</b> q0 à q9				0 ou 1	



Opérande	Signification		Gamme de valeurs
<b>v</b>	<b>Variables</b>		$\pm 2 \times 10^9$
	v0 à v100	Librement disponibles	
	v101	Variable de poursuite de position axe x1	
	v102	Variable de poursuite de position axe x2	
	v103	Variable de poursuite de position axe x3	
	v104	Variable de poursuite de position axe x4	
	v105 à v114	Réservé	
<b>x</b>	<b>Axe</b>		
	x1	pour WDP3-33X, WDP5-318, WP-311	
	x1 à x4	pour WPM-311	

Opérateurs possibles pour les instructions de déroulement:

Opérateur	Signification
"acc"	Sélectionner courbe d'accélération/décélération
"add"	Opération arithmétique addition (voir chapitre 3.4.1)
"and"	Opération logique: opération ET (voir chapitre 3.4.1)
"andn"	Opération logique: opération ET inversée (voir chapitre 3.4.1)
"cal"	Appeler sous-programme
"div"	Opération arithmétique division
"end"	Terminer programme de déroulement
"eq"	Opération de comparaison = (voir chapitre 3.4.1)
"gearn"	Définir facteur d'engrenage pour dénominateur
"gearz"	Définir facteur d'engrenage pour numérateur
"goff"	Déplacement de position par rapport à la grandeur de référence
"gt"	Opération de comparaison > (voir chapitre 3.4.1)
"handshake"	Synchronisation avec commande prioritaire
"jmp"	Saut au repère (voir chapitre 3.4.1)
"jmpc"	Saut conditionnel (voir chapitre 3.4.1)
"jmpn"	
"label"	Repère de saut (voir chapitre 3.4.1)
"ld"	Opération de transfert (voir chapitre 3.4.1)
"linmove" <sup>1)</sup>	Lancement d'une interpolation linéaire relative en unités définies par l'utilisateur
"linmovef" <sup>1)</sup>	Lancement d'une interpolation linéaire relative en unités définies par l'utilisateur et attendre jusqu'à ce que la position soit atteinte.
"linpos" <sup>1)</sup>	Lancement d'une interpolation linéaire absolue en unités définies par l'utilisateur

Opérateur	Signification
"linposf" <sup>1)</sup>	Lancement d'une interpolation linéaire absolue en unités définies par l'utilisateur et attendre jusqu'à ce que la position soit atteinte.
"lt"	Opération de transfert < (voir chapitre 3.4.1)
"mode"	Régler mode d'exploitation d'axe
"move"	Positionnement relatif en unités définies par l'utilisateur
"movef"	Positionnement relatif en unités définies par l'utilisateur et attendre jusqu'à ce que la position soit atteinte.
"mul"	Opération arithmétique multiplication
"nop"	Terminer programme (voir chapitre 3.4.1)
"or"	Opération logique: opération OU (voir chapitre 3.4.1)
"orn"	Opération logique: opération OU inversée (voir chapitre 3.4.1)
"pos"	Positionnement absolu en unités définies par l'utilisateur
"posf"	Positionnement absolu en unités définies par l'utilisateur et attendre jusqu'à ce que la position soit atteinte
"r"	Remettre à un opérande booléen (voir chapitre 3.4.1)
"rec_char"	Extraire caractères de la mémoire réceptrice
"rec_var"	Extraire variable de la mémoire réceptrice
"ref"	Réaliser course de référence vers interrupteur limiteur ou interrupteur de référence
"reff"	Réaliser course de référence et attendre jusqu'à ce que point de référence soit atteint
"ret"	Retour au programme principal
"s"	Mettre à un opérande booléen (voir chapitre 3.4.1)
"setipos" <sup>1)</sup>	Définition de la valeur de position en unités définies par l'utilisateur et préparation des axes qui doivent participer à l'interpolation linéaire.
"snd_str"	Ecrire chaîne de caractères dans la mémoire émettrice
"snd_var"	Ecrire variables dans la mémoire émettrice
"st"	Opération de transfert (voir chapitre 3.4.1)
"stop"	Arrêter déplacement d'axe ou interpolation linéaire
"stopa" <sup>1)</sup>	Arrêter déplacement des axes ou interpolation linéaire
"sub"	Opération arithmétique soustraction
"vel"	Régler vitesse exigée
"wait"	Instruction d'attente pour programme de déroulement

1) Ces instructions ne s'appliquent que pour les unités de positionnement multiaxes.

## Description des instructions de déroulement



### NOTE

Les instructions du programme d'édition de déroulement sont décrites ci-après. Les instructions API possibles sont décrites dans le chapitre 3.4.1.

### **acc**

---

Syntaxe:        `acc            <x><kv>`

L'instruction "acc" sert à introduire l'accélération maximale pour calculer la courbe d'accélération et de décélération sélectionnée (paramètre "Rampe" voir chapitre 3.3). Pendant les courses subséquentes, l'axe respectif x1 à x4 est accéléré et décéléré en utilisant cette rampe.

#### Exemple:

Opérateur	Opérande	Signification
ld	0	Charger valeur 0 dans ROL.
st	x1	Mémoriser ROL en tant que position de l'axe x1.
acc	x1    125	Rampe de l'axe x1 est accéléré ou décéléré avec 125 Hz/ms maxi.
:		

### **cal**

---

Syntaxe:        `cal            <l>`

L'instruction "cal" sert à appeler un sous-programme à partir d'un repère de saut prédéfini. Un maximum de trois sous-programmes peuvent être imbriqués l'un dans l'autre.

#### Exemple:

Opérateur	Opérande	Signification
:		
cal	l10	
:		
label	l10	Sauter au sous-programme à partir du repère de saut l10.
:		
ret		Retour depuis sous-programme.

### ***div***

---

Syntaxe:        `div`            `<kv>`

L'instruction "div" sert à réaliser une division avec le ROL et un opérande

*Exemple:*

Opérateur	Opérande	Signification
<code>ld</code>	<code>100</code>	Charger valeur 100 dans ROL.
<code>div</code>	<code>20</code>	Diviser le contenu du ROL (valeur 100) par la valeur 20.
<code>st</code>	<code>v4</code>	Mémoriser résultat en tant que variable v4.
:		

### ***end***

---

Syntaxe:        `end`

L'instruction "end" sert à terminer le programme de déroulement. Le repère de programme saute au début du programme et le programme recommence.

*Exemple:*

Opérateur	Opérande	Signification
<code>ld</code>	<code>0</code>	Charger valeur 0 dans ROL.
<code>st</code>	<code>x1</code>	Mémoriser ROL en tant que position de l'axe x1.
:		
<code>end</code>		Fin de programme et saut au début du programme.

### ***gearn***

---

Syntaxe:        `gearn`            `<x><kv>`

L'instruction "gearn" sert à régler le dénominateur du facteur d'engrenage pour les axes x1 à x4 en mode de poursuite de position. Ainsi, il est possible de réaliser un engrenage électronique. Il s'applique:

$$\text{Impulsions} = \text{unités d'encodeur} \times \frac{\text{gearn}}{\text{gearz}} + \text{goff}$$

Les unités d'encodeur dépendent du réglage du paramètre "évaluation d'encodeur E1 ou E2". Le facteur d'engrenage est pris en charge à l'aide de l'instruction "gearz".

*Exemple:*

Opérateur	Opérande	Signification
<code>gearn</code>	<code>x1 3</code>	Multiplier la valeur de position introduite sur l'entrée d'encodeur par le facteur d'engrenage 3/4.
<code>gearz</code>	<code>x1 4</code>	
<code>mode</code>	<code>x1 1</code>	Régler mode de poursuite de position via l'entrée d'encodeur.

## **gearz**

Syntaxe:        gearz        <x><kv>

L'instruction "gearz" sert à régler le numérateur du facteur d'engrenage pour les axes x1 à x4 en mode de poursuite de position. Ainsi, il est possible de réaliser un engrenage électronique. Il s'applique:

$$\text{Impulsions} = \text{unités d'encodeur} \times \frac{\text{gearn}}{\text{gearz}} + \text{goff}$$

Les unités d'encodeur dépendent du réglage du paramètre "évaluation d'encodeur E1 ou E2". Le facteur d'engrenage est pris en charge à l'aide de l'instruction "gearz".

*Exemple:*

Opérateur	Opérande	Signification
gearn	x1 3	Multiplier la valeur de position introduite sur l'entrée d'encodeur par le facteur d'engrenage 3/4.
gearz	x1 4	
mode	x1 1	Régler mode de poursuite de position via l'entrée d'encodeur.

## **goff**

Syntaxe:        goff        <x><kv>

L'instruction "goff" sert à ajuster les axes x1 à x4 avec une valeur de position en mode de poursuite de position (déplacement de position). Le déplacement de position est interprété comme position absolue et remis à zéro en changeant au mode de poursuite de position.

Si l'entraînement s'arrête ou l'on change à un autre mode d'exploitation, les impulsions introduites (positions) continuent à être saisies et déposées dans une mémoire intermédiaire. Si l'on retourne au mode de poursuite de position, ces impulsions sont extraites et un positionnement relatif de l'entraînement est réalisé.

La grandeur de référence (position) est prédéfinie par une entrée d'encodeur ou une variable en mode de poursuite de position (voir instruction "mode").

*Exemple:*

Opérateur	Opérande	Signification
gearn	x1 3	Facteur d'engrenage Régler 3/4.
gearz	x1 4	
mode	x1 1	Axe x1 est réglé en mode de poursuite de position via encodeur.
goff	x1 1000	Un déplacement de position de 1000 est additionné à la valeur de position actuelle de l'axe x1 et un positionnement est exécuté.
goff	x1 2000	De plus, l'axe est encore positionné de 1000 pas.

:

## ***handshake***

---

Syntaxe:        handshake    <i><q>

L'instruction "handshake" permet de programmer une synchronisation avec une commande prioritaire. Le programme utilisateur s'arrête et attend jusqu'à ce que l'entrée de synchronisation soit parcourue par le courant (= 1). Quand l'entrée est parcourue par le courant, la commande continue d'exécuter le programme utilisateur et la sortie de synchronisation est remise à zéro (se reporter au chapitre 1.11). Si l'entrée de synchronisation = 0, la sortie de synchronisation est mise = 1.

*Exemple:*

Opérateur	Opérande	Signification
:		
handshake	i10    q8	Le programme est arrêté jusqu'à ce que i10 = 1. Si le programme est à l'arrêt, la sortie q8 = 1, autrement 0.
:		

## ***linmove***

---

Syntaxe:        linmove

L'instruction "linmove" sert à lancer en mode point à point une interpolation linéaire préparée (voir instruction "setipos") vers une position relative en utilisant des unités définies par l'utilisateur. L'exécution de programme continue en parallèle. La valeur de position présélectionnée ne peut pas être modifiée pendant le positionnement.

*Exemple:*

Opérateur	Opérande	Signification
:		
setipos	x1    1000	Valeurs de position présélectionnées en unités définies par l'utilisateur pour les axes x1 à x3 qui participent à l'interpolation.
setipos	x2    2000	
setipos	x3    3000	
linmove		Lancement de l'interpolation linéaire vers une position relative.
:		

## ***linmovef***

---

Syntaxe:        `linmovef`

L'instruction "linmovef" permet de lancer en mode point à point une interpolation linéaire préparée (voir instruction "setipos") vers une position relative en utilisant des unités définies par l'utilisateur et d'attendre jusqu'à ce que cette position soit atteinte. L'exécution du programme est arrêtée pendant la course.

*Exemple:*

Opérateur	Opérande	Signification
:		
<code>setipos</code>	<code>x1 1000</code>	Valeurs de position
<code>setipos</code>	<code>x2 2000</code>	présélectionnées en unités définies par l'utilisateur pour les axes x1 à x2 qui participent à l'interpolation.
<code>linmovef</code>		Lancement de l'interpolation linéaire vers la position relative et attendre jusqu'à ce que cette position soit atteinte.
:		

## ***linpos***

---

Syntaxe:        `linpos`

L'instruction "linpos" sert à lancer en mode point à point une interpolation linéaire préparée (voir instruction "setipos") vers une position absolue en utilisant des unités définies par l'utilisateur. L'exécution de programme continue en parallèle. La valeur de position présélectionnée ne peut pas être modifiée pendant le positionnement.

*Exemple:*

Opérateur	Opérande	Signification
:		
<code>setipos</code>	<code>x1 5000</code>	Valeurs de position
<code>setipos</code>	<code>x2 4000</code>	présélectionnées en unités
<code>setipos</code>	<code>x3 3000</code>	définies par l'utilisateur pour les axes x1 à x3 qui participent à l'interpolation.
<code>linpos</code>		Lancement de l'interpolation linéaire vers la position absolue.
:		

## **linposf**

---

Syntaxe:        `linposf`

L'instruction "linposf" permet de lancer en mode point à point une interpolation linéaire préparée (voir "setipos") vers une position absolue en utilisant des unités définies par l'utilisateur et d'attendre jusqu'à ce que cette position soit atteinte. L'exécution du programme est arrêtée pendant la course.

*Exemple:*

Opérateur	Opérande	Signification
:		
setipos	x1    5000	Valeurs de position
setipos	x2    4000	présélectionnées en unités définies par l'utilisateur pour les axes x1 à x2 qui participent à l'interpolation.
linposf		Lancer l'interpolation linéaire vers la position absolue et attendre jusqu'à ce que la position soit atteinte.
:		

## **mode**

---

Syntaxe:        `mode            <x><k>`

L'instruction "mode" sert à régler le mode d'exploitation d'axe. Ce réglage ne peut être réalisé que si l'axe est à l'arrêt.

- 0    Mode point à point
- 1    Mode de poursuite de position via entrée d'encodeur. Les impulsions disponibles sur l'entrée d'encodeur sont converties en un mouvement du moteur.
- 2    Mode de poursuite de position au moyen d'une variable. La valeur dans la variable est convertie en un mouvement du moteur.



### **NOTE**

*Un facteur d'engrenage possible doit être réglé avant d'activer le mode de poursuite de position.*

Pour le mode de poursuite de position au moyen de variables, des variables de poursuite de position pour les axes x1 à x4 sont prédéfinies.

Variable	Signification
v101	Variable de poursuite de position axe x1
v102	Variable de poursuite de position axe x2
v103	Variable de poursuite de position axe x3
v104	Variable de poursuite de position axe x4



Le mode point à point est toujours réglé après la mise en circuit de la commande.

Les exemples suivants montrent la programmation des modes d'exploitation d'axe respectifs.

*Exemple 1:*

Opérateur	Opérande	Signification
:		
ld	0	Charger valeur 0 dans ROL.
st	x1	Mémoriser contenu du ROL en tant que valeur de position pour axe x1.
mode	x1 0	Axe x1 est réglé au mode point à point.
:		

*Exemple 2:*

Opérateur	Opérande	Signification
:		
gearn	x1 3	Facteur d'engrenage
gearz	x1 4	Régler 3/4.
mode	x1 1	Axe x1 est réglé au mode de poursuite de position au moyen de l'encodeur.
:		

*Exemple 3:*

Opérateur	Opérande	Signification
:		
gearn	x1 3	Facteur d'engrenage
gearz	x1 4	Régler 3/4.
mode	x1 2	Axe x1 est réglé au mode de poursuite de position au moyen de variables.
:		

## **move**

---

Syntaxe:            move                    <x><kv>

L'instruction "move" sert à présélectionner en mode point à point une position relative en utilisant des unités définies par l'utilisateur, à lancer un positionnement et à calculer une nouvelle position absolue. L'exécution de programme continue en parallèle. La valeur de position présélectionnée peut être modifiée pendant le positionnement. Pour calculer la nouvelle position, la valeur de position est additionnée à la position précédente.

Les états de mouvement (se déplace, à l'arrêt) du moteur respectif sont mémorisés dans des indicateurs pendant le positionnement des axes x1 à x4.

Indicateur	Signification	
	1	0
m1001	Axe x1 se déplace	Axe x1 est à l'arrêt
m1002	Axe x2 se déplace	Axe x2 est à l'arrêt
m1003	Axe x3 se déplace	Axe x3 est à l'arrêt
m1004	Axe x4 se déplace	Axe x4 est à l'arrêt

### *Exemple:*

Opérateur	Opérande	Signification
:		
move	x1      500	Une position relative de 500 unités définies par l'utilisateur est présélectionnée pour l'axe x1 et une nouvelle position absolue est calculée.
:		

## ***movef***

---

Syntaxe:        `movef        <x><kv>`

L'instruction "movef" permet de présélectionner en mode point à point une position relative en utilisant des unités définies par l'utilisateur, de lancer un positionnement et d'attendre jusqu'à ce que la position ait été atteinte. Une nouvelle position absolue est calculée en présélectionnant une position relative. L'exécution du programme est arrêté pendant la course.

Les états de signal des indicateurs m1001 à m1004 sont identiques à ceux-ci avec l'instruction "move".

### *Exemple:*

Opérateur	Opérande	Signification
:		
<code>movef</code>	<code>x1    1000</code>	Une position relative de 1000 unités définies par l'utilisateur est présélectionnée pour l'axe x1 et le programme attend jusqu'à ce que la nouvelle position absolue calculée ait été atteinte.
:		

## ***mul***

---

Syntaxe:        `mul        <kv>`

L'instruction "mul" permet de faire la multiplication avec un opérande et le ROL.

### *Exemple:*

Opérateur	Opérande	Signification
<code>ld</code>	<code>100</code>	Charger valeur 100 dans ROL.
<code>mul</code>	<code>20</code>	Multiplier ROL (valeur 100) par la valeur 20.
<code>st</code>	<code>v3</code>	Mémoriser résultat en tant que variable v3.
:		

### ***pos***

---

Syntaxe:        `pos`                    `<x><kv>`

L'instruction "pos" permet de présélectionner en mode point à point une position absolue en utilisant des unités définies par l'utilisateur et de lancer un positionnement. L'exécution de programme continue en parallèle. La valeur de position présélectionnée peut être modifiée pendant le positionnement.

Les états de signal des indicateurs m1001 à m1004 sont identiques à ceux-ci avec l'instruction "move".

*Exemple:*

Opérateur	Opérande	Signification
:		
ld	0	Charger valeur 0 dans ROL.
st	x1	Mémoriser ROL en tant que valeur de position pour axe x1.
pos	x1    1000	Une position absolue de 1000 unités définies par l'utilisateur est présélectionnée pour l'axe x1.
:		

### ***posf***

---

Syntaxe:        `posf`                    `<x><kv>`

L'instruction "posf" permet de présélectionner en mode point à point une position absolue en utilisant des unités définies par l'utilisateur et d'attendre jusqu'à ce que la position ait été atteinte. L'exécution du programme est arrêté pendant la course.

Les états de signal des indicateurs m1001 à m1004 sont identiques à ceux-ci avec l'instruction "move".

*Exemple:*

Opérateur	Opérande	Signification
:		
ld	0	Charger valeur 0 dans ROL.
st	x1	Mémoriser ROL en tant que valeur de position pour axe x1.
posf	x1    2000	Une position absolue de 2000 unités définies par l'utilisateur est présélectionnée pour l'axe x1 et on attend jusqu'à ce que la nouvelle position ait été atteinte.
:		

**rec\_char**

---

Syntaxe:        rec\_char        <v>

L'instruction "rec\_char" permet d'extraire un caractère de la mémoire réceptrice.

*Exemple:*

Opérateur	Opérande	Signification
:		
rec_char	v110	Caractère est reçu, p.ex. "A". La valeur ASCII du caractère est mémorisée dans la variable v110, p.ex. 65.
:		

**rec\_var**

---

Syntaxe:        rec\_var        <v>

L'instruction "rec\_var" permet d'extraire une variable de la mémoire réceptrice.

*Exemple:*

Opérateur	Opérande	Signification
:		
rec_var	v110	Variable v110 est reçue, p.ex. +219.
:		

**ref**

---

Syntaxe:        ref                <x>

L'instruction "ref" permet d'exécuter une course de référence vers un interrupteur limiteur ou de référence. L'interrupteur limiteur ou l'interrupteur de référence est sélectionné en modifiant le paramètre "type de la course de référence" (voir chapitre 3.3).

Une course de référence ne peut être exécutée qu'en mode point à point. Après avoir lancé la course de référence à l'aide de l'instruction "ref", le moteur se déplace à la vitesse réglée vers l'interrupteur limiteur ou l'interrupteur de référence. Ensuite, il s'approche du point de référence en sens inverse. Cette vitesse est réglée au moyen du paramètre "vitesse depuis interrupteur limiteur".

*Exemple:*

Opérateur	Opérande	Signification
:		
ref	x1	Le type respectif de la course de référence est exécuté.
:		

### ***reff***

---

Syntaxe:        `reff`            `<x>`

Comparée à l'instruction "ref", l'instruction "reff" ne présente qu'une seule différence. L'exécution du programme ne continue que si la course de référence a été réalisée avec succès.

Lorsqu'une erreur se produit pendant la course de référence, l'exécution du programme est bloquée par la course de référence.

*Exemple:*

Opérateur	Opérande	Signification
:		
<code>reff</code>	<code>x1</code>	Une course de référence est exécutée et le programme attend jusqu'à ce que le point de référence ait été atteint.
:		

### ***ret***

---

Syntaxe:        `ret`

L'instruction "ret" sert à abandonner un sous-programme.

### ***setipos***

---

Syntaxe:        `setipos`        `<x><kv>`

L'instruction "setipos" permet de préparer en mode point à point une interpolation linéaire avec deux ou trois axes. Une position en unités définies par l'utilisateur doit être présélectionnée pour chaque axe qui participe à l'interpolation.

Le nombre des instructions "setipos" successives définit si l'interpolation se fait avec 2 ou 3 axes.

L'interpolation est lancée au moyen d'une instruction (p.ex. "linmove" ou "linpos").

La valeur de position présélectionnée ne doit pas être modifiée pendant le positionnement.

*Exemple 1:*

Opérateur	Opérande	Signification
:		
<code>setipos</code>	<code>x1</code> <code>1000</code>	Présélection des valeurs de position en unités définies par l'utilisateur pour les axes <code>x1</code> à <code>x3</code> qui participent à l'interpolation.
<code>setipos</code>	<code>x2</code> <code>2000</code>	
<code>setipos</code>	<code>x3</code> <code>3000</code>	
<code>linmove</code>		Lancer l'interpolation linéaire vers une position relative.
:		

*Exemple 2:*

Opérateur	Opérande	Signification
:		
setipos	x1 1000	Présélection des valeurs de position en unités définies par l'utilisateur pour les axes x1 à x3 qui participent à l'interpolation.
setipos	x2 2000	
setipos	x3 3000	
linpos		Lancer l'interpolation linéaire vers la position absolue.
:		

***snd\_str***

---

Syntaxe:        `snd_str`        <kv>

L'instruction "snd\_str" permet d'écrire une ligne de texte (40 caractères maxi) dans le tampon émetteur et de la sortir automatiquement sur l'interface qui a été sélectionnée au moyen du paramètre "unité de commande".

La ligne de texte doit être introduite à l'aide du programme d'édition de texte, p.ex.:

```
14: Une course de référence est exécutée
:
```

*Exemple:*

Opérateur	Opérande	Signification
:		
snd_str	14	La ligne de texte 14 "Une course de référence est exécutée" est sortie via l'interface sérielle.
:		
ld	14	
st	v1	
snd_str	v1	La ligne de texte 14 "Une course de référence est exécutée" est sortie via l'interface sérielle.

### **snd\_var**

---

Syntaxe:        `snd_var`        `<v>`

L'instruction "snd\_var" permet d'écrire une variable dans le tampon émetteur.

Une variable, p.ex. la variable de poursuite de position v101 de l'axe x1, est sortie dans le programme de déroulement via l'interface sérielle.

*Exemple:*

Opérateur	Opérande	Signification
:		
snd_var	v101	Valeur de la variable de poursuite de position v101 pour axe x1, p.ex. +219111 est sortie.
:		

### **stop**

---

Syntaxe:        `stop`        `<x>`

L'instruction "stop" permet d'arrêter un déplacement d'axe ou une interpolation linéaire. Après avoir introduit l'instruction "stop", l'entraînement devient inactif et peut être redémarré au moyen de l'instruction "pos".

*Exemple:*

Opérateur	Opérande	Signification
:		
stop	x1	Axe x1 est arrêté.
:		

### **stopa**

---

Syntaxe:        `stopa`

L'instruction "stopa" permet d'arrêter tous les déplacements d'axes ou une interpolation linéaire. Après avoir introduit l'instruction "stopa", les entraînements deviennent inactifs et peuvent être redémarrés au moyen de l'instruction "pos".

Une interpolation linéaire vers une position présélectionnée qui a été interrompue peut être terminée en redémarrant les entraînements.

*Exemple:*

Opérateur	Opérande	Signification
:		
stopa		Tous les déplacements d'axes ou une interpolation linéaire sont arrêtés.
:		



## **vel**

---

Syntaxe:        vel                    <x><kv>

L'instruction "vel" permet de régler la vitesse exigée d'un axe. Lorsqu'une vitesse exigée n'est pas réglée en mode point à point, l'axe se déplace à la "vitesse standard" qui est introduite dans le programme d'édition de paramètres.

En mode point à point, il est possible de modifier la vitesse exigée avant ou pendant le positionnement. La vitesse exigée en mode point à point doit toujours être supérieure à 0.

*Exemple:*

Opérateur	Opérande	Signification
:		
vel	x1 10000	Une vitesse exigée de 10000 Hz est réglée pour l'axe x1.
:		

## **wait**

---

Syntaxe:        wait                    <kv>

L'instruction "wait" définit un temps d'attente dans le programme de déroulement. La résolution de temps est de 1 ms.

Si wait = 0, la puissance de calcul jusqu'au prochain rythme de 2 ms est attribuée au programme API. Ce réglage n'est raisonnable que si p.ex. l'interrogation d'une entrée est programmée en tant que boucle dans le programme de déroulement. Le temps de traitement nécessaire à l'interrogation de l'entrée est p.ex. de 0,01 ms. Dans ce cas, le programme API dispose du temps de calcul résiduel de 1,99 ms.

*Exemple 1:*

Opérateur	Opérande	Signification
:		
wait	10	Le programme de déroulement est arrêté pendant 10 ms.
:		

*Exemple 2:*

Opérateur	Opérande	Signification
label	l1	
ld	i1	Tant que l'entrée i1 = 0, le temps de calcul complet est attribué au programme API.
jmpc	l2	
:		
wait	0	
jmp	l1	
label	l2	

## 3.5.2 Afficher programme de déroulement

Afficher le programme de déroulement  
Afficher le segment désiré

<L> <↓>  
<L1-14> <↓>

```
DER-Edit> l1-14

1:  ld      0
2:  st      x1
3:  gearn   x1      1000
4:  gearz   x1      1000
5:  mode    x1      1
6:  label   l1
7:  ld      i10
8:  jmpn    -1
9:  mode    x1      0
10: ld      0
11: st      x1
12: vel     x1      15000
13: movef   x1      10000
14: end

DER-Edit> _
```

*Fig. 3-12 Exemple d'un programme de déroulement*

## 3.5.3 Insérer une ligne

Insérer une ligne, p.ex. à partir de la ligne 12  
Insérer et terminer l'édition

<I12> <↓>  
<ESC>

Exemple:

```
DER-Edit>I12 <↓>
DER-Edit-I0012>LABEL l2<↓>
DER-Edit-I0013>ADD 100 <↓>
DER-Edit-I0014> <ESC>
```

### 3.5.4 Effacer ligne(s)

Effacer ligne, p.ex. ligne 5

<D5> <↵>

Effacer lignes, p.ex. lignes 1 à 5

<D1-5> <↵>



**NOTE**

*L'opération d'effacement est immédiatement exécutée et le programme de déroulement complet mis à jour.*

Exemple:

```
DER-Edit>D5 <↵>
```



**NOTE**

*Après avoir introduit <↵> OK est affiché.*

### 3.5.5 Editer une ligne

Appeler la ligne désirée, p.ex. ligne 11

<E11> <↵>

Terminer l'édition

<ESC>

Exemple:

```
DER-Edit>E11 <↵>
DER-Edit-0011>vel x1 15000 <↵>
DER-Edit-0012>vel x1 20000 <↵>
DER-Edit-0013>movef x1 10000 <↵>
DER-Edit-0014> <ESC>
```

## 3.6 Programme d'édition de texte

Le programme d'édition de texte sert à définir des textes qui sont visualisés, avec les instructions respectives, dans le programme de déroulement sur une unité de sortie (p.ex. FT 2000).

Le numéro de ligne du texte à afficher est spécifié comme paramètre dans l'instruction OED3 qui est prévue pour la sortie de la chaîne de caractères.

Sélectionner le programme d'édition de texte	<T>	<↵>
Appeler la sélection de fonctions	<H>	<↵>

Les lignes et segments suivants sont indiqués à titre d'exemple.

### 3.6.1 Afficher textes

Afficher textes	<L>	<↵>
Afficher le segment désiré	<L1-14>	<↵>

```
TXT-Edit> l1-14

1:  Commentaire
2:  Vitesse
3:  STRING3
4:  STRING4
5:  Course de référence
6:  STRING6
7:  STRING7
8:  STRING8
9:  Mode point à point
10: STRING10
11: STRING11
12: STRING12
13: STRING13
14: STRING14

TXT-Edit> _
```

Fig. 3-13 Exemples de textes

### 3.6.2 Editer textes

Appeler la ligne désirée, p.ex. ligne 3	<E3>	<↵>
Terminer l'édition	<ESC>	

Exemple:

```
TXT-Edit>E3
TXT-Edit-0003>STRING3
TXT-Edit-0004>Mode de poursuite de position
TXT-Edit-0005>Course de référence
TXT-Edit-0006> <ESC>
```



#### NOTE

Un maximum de 48 lignes de 40 caractères chacune peuvent être introduites.

## 3.7 Fonction Upload

La fonction Upload permet d'extraire toutes les données d'utilisateur (p.ex. programme de déroulement, programme API, textes, paramètres) de la commande, de les afficher via l'interface sur l'écran et de les mémoriser dans un fichier.

Lorsque le programme de terminal BTERM est utilisé, la fonction Upload est exécutée de la manière suivante:

1. La commande est en mode d'édition.
2. Appuyer sur la touche <F8>.  
→ Les options "ouvrir" et "fermer" sont affichées dans une fenêtre.
3. Confirmer l'option "ouvrir" en appuyant sur <↓>.  
→ La fenêtre "nom de fichier" apparaît.
4. Introduire le nom de fichier désiré et confirmer avec <↓>.  
→ Le programme de terminal BTERM ajoute l'extension .CAP, p.ex. TEST.CAP.
5. Lancer la fonction Upload: <U> <↓>  
→ Toutes les données d'utilisateur comprenant le programme de déroulement, le programme API, des textes et des paramètres sont extraites de la commande et affichées sur l'écran.
6. Après l'extraction de toutes les données d'utilisateur, appuyer sur la touche <F8>.
7. Confirmer l'option "fermer" avec <↓>.  
→ Les données d'utilisateur sont mémorisées dans le fichier TEST.CAP.



### NOTE

*Le fichier TEST.CAP peut être traité au moyen d'un programme d'édition de texte facultatif. De plus, il est possible d'introduire des commentaires ligne par ligne. Un commentaire doit toujours commencer au début d'une ligne par un point-virgule, p.ex. c'est un commentaire. Les commentaires ne sont pas transmis dans la commande en exécutant la fonction Download.*

### 3.8 Fonction Download

La fonction Download sert à charger les données qui ont été sauvegardées au moyen de la fonction Upload dans la commande et à les soumettre à un contrôle de validité.

Lorsqu'une erreur se produit, l'opération de chargement dans la commande est interrompue et les données sont rebutées.

Lorsque le programme de terminal BTERM est utilisé, la fonction Download est exécutée de la manière suivante:

1. La commande est en mode d'édition.
2. Les données d'utilisateur sont sauvegardées dans un fichier, p.ex. TEST.CAP. Changer le nom de l'extension de .CAP en .MAC (vous trouverez les instructions pour changer le nom de l'extension dans le manuel MS-DOS).
3. Lancer la fonction Download: <D> <↵>
4. Appuyer sur la touche <F6>.  
→ Les options "ouvrir", "fermer" et "exécuter" apparaissent dans une fenêtre.
5. Confirmer l'option "exécuter" avec <↵>.  
→ Une fenêtre qui comprend les fichiers .MAC apparaît sur l'écran.
6. Sélectionner le fichier désiré, p.ex. TEST.MAC.  
→ Les données d'utilisateur sont chargées dans la commande et soumises à un contrôle de validité.

### 3.9 Terminer le mode d'édition

La surface de programmation OED3 est abandonnée avec <X> <↵>.



#### **ATTENTION**

***Le programme utilisateur introduit est exécuté aussitôt que le mode d'édition a été terminé.***

## 4 Messages d'erreur

Les erreurs décrites dans le tableau suivant peuvent se produire pendant la génération et l'exécution du programme. Ces erreurs sont sorties via l'interface sérielle.

Type d'erreur/Numéro d'erreur	Cause/Élimination
<p>Erreur de syntaxe lors de la génération du programme</p> <p>[E4291] [E4295] [E4296] [E4297] [E4299] [E4300] [E4302]</p>	<p>Des erreurs de syntaxe sont immédiatement affichées sur l'écran et doivent être corrigées, ainsi assurant qu'il n'y a pas d'erreurs de syntaxe dans le programme après avoir abandonné le programme d'édition.</p> <p>Commande non réalisée Instruction inadmissible dans le programme API Opérande est incorrect Instruction inconnue Erreur de système OED3 Erreur à l'introduction numérique Conflit de données à l'appel de la OED3</p>
<p>Erreurs au paramétrage</p> <p>[E4290] [E4292] [E4293] [E4294] [E4298]</p>	<p>Les gammes de validité des valeurs introduites sont immédiatement contrôlées. Lorsque des valeurs incorrectes ont été introduites, les limites sont affichées et une nouvelle introduction est attendue.</p> <p>Paramétrage incomplet OED3 Paramètres incorrects Nombre insuffisant de paramètres Nombre excessif de paramètres Paramétrage incorrect</p>
<p>Erreurs pendant transmission de données (Download) dans la commande</p>	<p>La fonction Download contrôle la validité de chaque ligne qui a été transmise dans la commande, cela signifie que la gamme de validité des paramètres ainsi que la syntaxe des lignes de programme sont contrôlés.</p> <p>Lorsqu'une erreur se produit, la transmission dans la commande est interrompue. Toutes les données qui sont transmises après l'erreur sont rebutées.</p> <p>L'erreur doit être confirmée en appuyant sur &lt;ESC&gt;.</p>
<p>Erreurs pendant l'exécution du programme (se reporter aux pages suivantes pour la sortie d'erreurs)</p>	<p>Des erreurs qui se produisent pendant l'exécution de programme sont signalées par un nombre clignotant sur l'indicateur d'état 40 et inscrites dans la mémoire d'erreurs de la commande (se reporter à la documentation du dispositif).</p> <p>De plus, la sortie d'erreur de la commande apparaît sur l'écran (voir figure 4-1).</p> <p>Des erreurs de système (erreurs fatales) de la commande entraînent la remise à zéro de la commande et sont affichées sur l'indicateur d'état 40 (se reporter à la documentation du dispositif).</p>

## Messages d'erreur

<pre> Erreur de commande INIT: xxxx Déroulement: xxxx API: xxxx </pre>
<pre> Encodeur 1: xxxx Encodeur 2: xxxx Serial c1: xxxx Serial c2: xxxx </pre>
<pre> Entrées i: xxxxxxxxxxxxxxxxxxxxxxxx Sorties q: xxxxxxxxxxxx </pre>
<pre> Axe x1: xxxx Signaux x1: xxxx Axe x2: xxxx Signaux x2: xxxx </pre>
<pre> Axe x3: xxxx Signaux x3: xxxx Axe x4: xxxx Signaux x4: xxxx &lt;cr&gt; </pre>
<p>□ pour affichage de deux lignes</p>

Fig. 4-1 Erreurs visualisées sur l'écran

Le tableau suivant explique le contenu de la mémoire d'erreurs et la signification de la sortie d'erreur montrée dans la figure 4-1.

Si plusieurs d'erreurs d'un groupe d'erreurs se produisent simultanément, le total des numéros d'erreur individuels est sorti.

La documentation du dispositif comporte des instructions relatives à l'élimination d'erreurs.

Sortie d'erreur	Signification
Erreur de commande	INIT: Numéro de ligne dans le programme INIT
	Déroulement: Numéro de ligne dans le programme de déroulement
	API: Numéro de ligne dans le programme API
Encodeur 1 ou 2	
16#0100	Erreur de poursuite
16#0200	Erreur d'encodeur
Serial c1 ou c2	
16#0002	Accès à une interface non initialisée
16#0010	Interface de matériel déjà affectée
16#0020	Erreur lors de l'affectation ou désaffectation de la mémoire
16#0040	Mémoire tampon réceptrice insuffisante pour chaîne de caractères
16#0080	Chaîne de caractères transmise excessive
16#0100	Suite de caractères inadmissible dans chaîne de caractères transmise
16#0800	Débordement dans mémoire réceptrice de communication
16#1000	Overrun: vitesse baud excessive
16#2000	Parity: erreur de transmission
16#4000	Framing: paramètres d'interface incorrects
16#7FFF	Break: discontinuité de câble ou communication n'est pas disponible



Sortie d'erreur	Signification
<p>Axes x1 à x4</p> <p>16#0001 16#0004 16#0008 16#0010 16#0020 16#0040 16#0080 16#0100 16#0400 16#0800 16#1000 16#2000 16#4000 16#7FFF</p>	<p>Un ou plusieurs signaux actifs, voir "signaux x1 à x4"</p> <p>Instruction inadmissible en mode de poursuite de position</p> <p>Positionnement d'attente déjà actif (p.ex. "posf", "movef")</p> <p>Instruction inadmissible si l'axe est interrompu/bloqué</p> <p>Courbe maîtresse incorrecte</p> <p>Informations insuffisantes sur la grandeur de référence</p> <p>Instruction inadmissible pendant déplacement d'axe</p> <p>Instruction inadmissible pendant course de référence</p> <p>Valeur incorrecte pour paramètre</p> <p>Une valeur ne peut pas être calculée</p> <p>Transmission d'une valeur incorrecte pour un paramètre</p> <p>Instruction ne peut pas être exécutée sous ces conditions</p> <p>Interrogation d'une valeur non définie</p> <p>Ressource non prêt</p>
<p>Signaux x1 à x4</p> <p>16#0001 16#0002 16#0004 16#0008 16#0010 16#0020 16#0040 16#0080 16#0100 16#0200 16#0400 16#0800 16#1000</p>	<p>Interrupteur limiteur de matériel positif</p> <p>Interrupteur limiteur de matériel négatif</p> <p>Interrupteur de référence</p> <p>Arrêt matériel</p> <p>Trigger matériel</p> <p>Interrupteur limiteur de logiciel positif</p> <p>Interrupteur limiteur de logiciel négatif</p> <p>Arrêt logiciel</p> <p>Erreur de poursuite</p> <p>Erreur d'encodeur</p> <p>Unité de puissance n'est pas "prête"</p> <p>Echauffement unité de puissance</p> <p>Echauffement moteur</p>



**NOTE**

*Les erreurs inscrites dans la mémoire d'erreurs sont effacées après la remise à zéro de la commande ou après le redémarrage du programme utilisateur.*



**NOTE**

*Dépendant du réglage du paramètre "unité de commande", la sortie d'erreur peut comprendre deux ou plusieurs lignes, se reporter au chapitre 1.10.*



## 5 Annexe

### 5.1 Termes techniques

#### *Accumulateur*

Registre dans la commande qui est utilisé pour la mise en mémoire de résultats intermédiaires. Le contenu de cet accumulateur s'appelle également résultat d'opération logique (ROL).

#### *Course de référence*

Les instructions "ref" et "reff" sont utilisées pour réaliser des courses de référence. Une course de référence n'est possible qu'en mode point à point. Pendant la course de référence on s'approche d'un point de référence qui doit être le zéro du système de coordonnées.

#### *Encodeur*

Capteur pour la détection de position (détection de la position réelle) d'un moteur.

#### *Engrenage électronique*

En mode de poursuite de position, la multiplication ou démultiplication de la grandeur de référence (p.ex. encodeur) est possible grâce à un facteur d'engrenage. Cela s'appelle également engrenage électronique. Il s'applique la relation suivante:  
unités d'entraîn. = grandeur de référence x facteur d'engrenage

#### *Entrées/sorties*

La commande dispose d'un nombre fixe d'entrées/sorties. Ces entrées/sorties sont utilisées pour commander le déroulement. Le traitement des entrées/sorties et l'exécution de déplacements/mouvements peuvent être réalisés en parallèle.

#### *Evaluation d'encodeur*

Ce paramètre sert à régler la résolution d'encodeur (500 ou 1000 traits) et le facteur d'évaluation interne (simple, double ou quadruple). Il s'applique la relation suivante:

$$\text{Révol. du mot.} = \text{impulsions d'encodeur} \times \frac{\text{facteur d'évaluation}}{\text{résolution d'encodeur}}$$

#### *Facteur de cadrage*

Le facteur de cadrage sert à convertir des unités définies par l'utilisateur (p.ex. cm) en unités d'entraînement (pas ou incréments).

#### *Image de processus PA*

L'image de processus sert à la mise en mémoire intermédiaire des entrées/sorties.

#### *Indicateurs*

Les indicateurs sont des places en mémoire qui peuvent être utilisées. La commande dispose d'une zone de mémoire spécifique pour les indicateurs.

#### *Interpolation*

Course coordonnée simultanée de plusieurs axes (deux axes au minimum).

### *Mode de poursuite de position*

En mode de poursuite de position, les positions sont présélectionnées au moyen d'une entrée d'encodeur ou d'une variable. La position peut être ajustée au moyen d'un facteur d'engrenage. Ainsi, il est possible de réaliser un engrenage électronique.

### *Mode point à point*

En mode point à point, un déplacement d'un point A à un point B est exécuté à l'aide d'une instruction de positionnement. Le positionnement peut être absolu (basé sur le point de référence de l'axe) ou relatif (basé sur la position actuelle de l'axe).

### *Opérandes*

Beaucoup d'instructions demandent un opérande qui est utilisé pour réaliser l'opération logique.

### *Opérations arithmétiques*

Les instructions "add", "sub", "mul", "div" permettent de réaliser des opérations arithmétiques avec un opérande et le ROL.

### *Opérations de comparaison*

Les instructions "eq", "gt" et "lt" permettent de faire une comparaison avec un opérande et le ROL. Après la comparaison, le ROL est toujours du type de données BOOLEEN et comporte les valeurs:

0 ou FAUX, si la comparaison était fautive ou  
1 ou VRAI, si la comparaison était vraie.

### *Opérations de transfert*

Les instructions "ld" et "st" permettent de charger des valeurs dans le ROL et de les mémoriser depuis le ROL dans des variables. Lors du chargement "ld", le ROL adopte le type de données de la valeur chargée. Lors de la mise en mémoire "st", les types de données du ROL et de l'opérande doivent être compatibles.

### *Opérations logiques*

Les instructions "and" et "or" permettent de réaliser des opérations logiques avec un ou plusieurs opérandes booléens et le ROL.

### *Position exigée*

En mode point à point, les positions exigées sont présélectionnées au moyen des instructions "pos(f)" ou "move(f)" et un positionnement est déclenché.

### *Positionner et repositionner*

Les instructions positionner "s" et repositionner "r" permettent de positionner une variable BOOLEENNE (p.ex. sortie ou indicateur) à 1 ou de la repositionner à 0 en fonction du résultat d'opération logique.

### *Programmation d'encodeur*

Chaque entrée d'encodeur peut être utilisée pour l'introduction de grandeurs de référence (en mode de poursuite de position) ou pour le contrôle de rotation.

*Résultat d'opération logique ROL*

Le résultat d'opération logique ROL est une mémoire intermédiaire (accumulateur) de la commande qui est utilisée pour des opérations arithmétiques, des opérations logiques et la transmission de données. A la différence d'automates programmables industriels usuels, une commande BERGER LAHR de la série 300 dispose d'un seul accumulateur pour le résultat d'opération logique. La capacité de mémoire de l'accumulateur est automatiquement adaptée au type de données de l'opérande.

*Unités d'entraînement*

Les unités d'entraînement sont des opérandes internes de la commande pour le calcul de positions, vitesses et accélérations.

*Unités définies par l'utilisateur*

Les unités définies par l'utilisateur sont des opérandes qui peuvent être facultativement déterminés. Il s'applique la relation suivante:  
unités d'entraînement = unités définies par l'utilisateur x facteur de cadrage  
(unités d'entraînement en pas ou incréments, unités définies par l'utilisateur p.ex. en cm)  
Des positions, vitesses et accélérations sont toujours spécifiées en unités définies par l'utilisateur.

*Variables*

Des variables sont des places en mémoire qui sont utilisées pour l'échange de données et la mise en mémoire de données dans un programme.

### 5.2 Abréviations

add	Instruction d'addition
and	Opération ET
API	Automate programmable industriel
ASCII	American Standard Code for Information Interchange
B	Byte, octet
b	Valeur booléenne
c1	Interface sérielle 1
c2	Interface sérielle 2
c.a.	Tension alternative
c.c.	Tension continue
cal	Appel d'un sous-programme
div	Instruction de division
Doc. No.	Numéro de documentation
E	Encodeur
eq	Instruction de comparaison, égal
gt	Instruction de comparaison, supérieur à
Hz	Hertz
i	Entrée
jmp	Saut à un repère

k	Valeur
l	Label
ld	Charger
LIMN	Interrupteur limiteur négatif
LIMP	Interrupteur limiteur positif
lt	Instruction de comparaison, inférieur à
m	Indicateur
M	Moteur
ms	Milliseconde
mul	Instruction de multiplication
N	Nombre de traits de l'encodeur
or	Opération OU
PA	Image de processus
PID	Régulateur proportionnel, intégral, différentiel
q	Sortie
r	Repositionner
ref	Course de référence
ret	Retour au sous-programme
ROL	Résultat d'opération logique

## Annexe

---

s	Positionner
st	Mémoriser
Stop	Signal d'arrêt
sub	Instruction de soustraction
t	Timer
TRIG	Signal trigger
v	Variable
x	Numéro d'axe



**6 Index****A**

Accessoires	2-1
Axe	3-11, 3-27

**C**

Câble d'interface	2-1
Carte d'entrée/sortie MP 926	1-22
Conditions de matériel	2-2
Constantes	3-11, 3-26
Contrôle de rotation	1-15
Convertisseur d'interface	2-1
Course de référence	1-10

**D**

Déplacement de position	1-14
-------------------------	------

**E**

Entrées	3-11, 3-26
---------	------------

**I**

Indicateurs	1-6, 3-11, 3-26
Instruction	
acc	3-29
add	3-13
and	3-13
cal	3-29
div	3-14, 3-30
end	3-14, 3-30
eq	3-15
gearn	3-30
gearz	3-31
goff	3-31
gt	3-15
handshake	3-32
jmp	3-16
label	3-17
ld	3-17

Instruction	
linmove	3-32
linmovef	3-33
linpos	3-33
linposf	3-34
lt	3-18
mode	3-34
move	3-36
movef	3-37
mul	3-37
nop	3-19
or	3-19
pos	3-38
posf	3-38
r	3-20
rec_char	3-39
rec_var	3-39
ref	3-39
reff	3-40
ret	3-40
s	3-20
setipos	3-40
snd_str	3-41
snd_var	3-42
st	3-21
stimer	3-21
stop	3-42
stopa	3-42
sub	3-22
vel	3-43
wait	3-43
Interpolation linéaire	1-16
<b>L</b>	
Label	3-11, 3-26
Lancer programme	3-1
Liste de paramètres	3-6

**M**

Messages d'erreur	4-1 - 4-4
Mode automatique	1-6
mode de poursuite de position	1-11
Présélection de la valeur de position	1-11
Mode manuel	1-6
Mode pas à pas	1-7
Mode point à point	1-8
Mode teach-in	1-7
Modes d'exploitation	1-5

**O**

Opérandes	3-11, 3-26
-----------	------------

**P**

Paramètres	
afficher	3-5
éditer	3-9
Point de référence	1-10
Positionnement	
absolu	1-8
relatif	1-8
Programmation d'interface	1-19
Programmation de déplacement	1-8
Programmation de l'encodeur	1-13
Programme API	
Appeler liste d'instructions	3-10
Editer ligne	3-24
Effacer ligne(s)	3-24
Insérer ligne	3-23
Liste d'instructions	3-10
Programme d'édition API	3-10
Programme d'édition de déroulement	3-25
Programme d'édition de paramètres	3-5
Programme d'édition de texte	3-46
Afficher textes	3-46
Editer textes	3-46

Programme de déroulement	
Afficher programme	3-44
Appeler liste d'instructions	3-25
Editer ligne	3-45
Effacer ligne(s)	3-45
Insérer ligne	3-44
<b>R</b>	
Régulation de position	1-18
Résultat d'opération logique ROL	3-12
<b>S</b>	
Sauvegarde des données	
Download	3-48
EEPROM	3-3
Upload	3-47
Schéma de connexion	2-2
Sorties	3-11, 3-26
Surface de programmation	3-3
<b>T</b>	
Terminal de commande FT 2000	1-19
Terminer programme	3-48
Timer	3-11
<b>V</b>	
Valeur booléenne	3-11, 3-26
Variables	3-11, 3-27
Volume de livraison	2-1
<b>Z</b>	
Zéro	1-10

**BERGER LAHR GmbH**

Breslauer Str. 7  
Postfach 1180

D-77901 Lahr

**Proposition  
Corrections**

**OED3**

Edition: a000 Avril 94  
Doc. No. 212.954/DF 04.94

**Exéditeur:**

Nom:

Etablissement/Service:

Adresse:

Téléphone:

Si vous avez constaté des erreurs, nous vous prions de nous les faire savoir en utilisant ce formulaire.

En outre, nous vous serions reconnaissants de nous envoyer vos propositions et suggestions.

**Propositions et/ou corrections:**